(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 947 954 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
06.10.1999 Bulletin 1999/40

(51) Int Cl.6: G06T 3/00

(21) Application number: 99301849.8

(22) Date of filing: 11.03.1999

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 30.03.1998 US 52039

(71) Applicant: SEIKO EPSON CORPORATION
Shinjuku-ku Tokyo (JP)

(72) Inventors:
- Ratnakar, Viresh
San Jose, California 95134-1627 (US)
- Ivashin, Victor
San Jose, California 95134-1627 (US)
- Bhaskaran, Vasudev
San Jose, California 95134-1627 (US)

(74) Representative: Sturt, Clifford Mark et al
Miller Sturt Kenyon
9 John Street
London WC1N 2ES (GB)

(54) **Image transformations in the compressed domain**

(57) Image processing techniques which involve direct manipulation of the compressed domain representation of an image to achieve the desired spatial domain processing without having to go through a complete decompression and compression process. The techniques include processing approaches for performing the eight operations in D_4 (the dihedral group of symmetries of a square) on JPEG images using the discrete cosine transform (DCT) domain representation of the images directly. For a task such as image rotation by 90° (an operation in D_4), DCT-domain based methods can yield nearly a five-fold increase in speed over a spatial-domain based technique. These simple compressed-domain based processing techniques are well suited to the imaging tasks that are needed in a JPEG-based digital still-camera system.

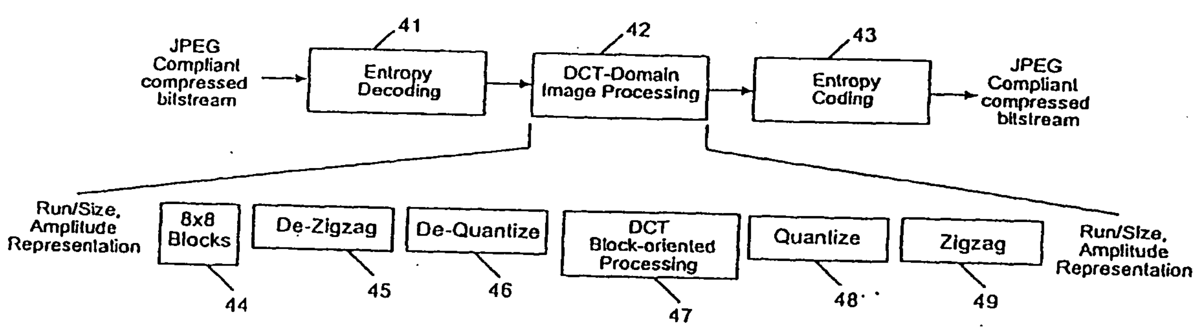

Fig. 3

EP 0 947 954 A1

Description

[0001] This invention relates generally to the processing of the compressed domain representation of image data, and more particularly to the manipulation of the compressed domain representation to achieve certain spatial domain processing, such as regular geometric transformations of an image, without having to subject the image data to the full decompression and compression process.

[0002] A typical high quality digitized color image may use 24 bits per pixel (bpp)--8 bits each for red (R), green (G) and blue (B) in RGB color space or for luminance (Y), chrominance (C_B) and chrominance (C_R) in YC_BC_R color space. To transmit or store such images in the uncompressed state (i.e., in spatial or pixel domain) is simply too costly in terms of time and memory requirements. Thus, applications and devices which store and/or transmit high quality digitized color images, such as digital cameras, typically do so in a compressed format, using one of the currently available compression algorithms.

[0003] The emergence of compression standards such as JPEG (an acronym for "Joint Photographic Experts Group") has led to many digital imaging systems and applications that create and maintain content only in JPEG compressed format. For instance, in most digital still-imaging cameras (DSCs) such as the Epson PhotoPC 600, Kodak DC-10, etc., pictures captured by the camera are immediately compressed within the camera and stored in the camera's storage system as JPEG files. Often there is a need to manipulate these pictures prior to display. Typical image manipulations might include (a) rotating the picture from portrait to landscape mode and vice-versa, (b) scaling the picture to increase or decrease its size, (c) changing brightness and contrast in the picture, (d) cropping portions of the picture for the purposes of creating a new picture and for compositing operations, (e) adding simple bitmap annotations to a picture, and (f) embedding visible/invisible watermarks in the picture. Due to storage constraints within the digital camera, these image manipulations require the processed output to be in JPEG format.

[0004] The need to do these tasks and the availability of the picture only in the compressed mode has resulted in a great deal of interest in developing image processing techniques that can be applied directly to the compressed domain representation. The motivation for investigating compressed domain processing methods stems from the observations that (a) the volume of data in compressed domain tends to be quite small compared to the spatial domain representation which means that fewer operations per sample may be required for the desired image processing task, and (b) conventional processing pipelines that require the data to be decompressed, followed by the application of the desired image processing function in spatial domain, and then recompressed for transmission or storage efficiency can lead to a loss in image fidelity. Furthermore, such a conventional processing pipeline has very high computation complexity or high latency since the compression task is often more complex than the decompression task. The compressed-domain based processing methodology on the other hand, often leads to reduced computation complexity since it replaces the JPEG decompression and compression tasks by low complexity tasks such as Huffman decoding and Huffman encoding. (See, S. F. Chang and D. G. Messerschmitt, "Manipulation and Compositing of MC-DCT Compressed Video," *IEEE JSAC Special Issue on Intelligent Signal Processing,* vol. 13, no. 1, pp. 1-11, Jan. 1995; N. Merhav and V. Bhaskaran, "A fast algorithm for DCT-domain inverse motion compensation," *Proc. ICASSP '96,* pp. IV. 2307-2310, Atlanta, May 1996; B. Natarajan and V. Bhaskaran, "A fast approximate algorithm for scaling down digital images in the DCT domain," *IEEE International Conference on Image Processing (ICIP),* Washington, D.C., Oct. 1995; and Brian Smith and Larry Rowe, "Algorithms for manipulating compressed images," *IEEE Computer Graphics and Applications,* pp. 34-42, Sept. 1993.)

[0005] Therefore, it is an object of the present invention to overcome the aforementioned problems associated with performing manipulations on digital images in spatial domain.

[0006] It is another object of this invention to provide for direct manipulation of the compressed domain representation of an image to bring about a selected spatial domain manipulation without having to go through a complete decompression and compression process.

[0007] It is a further object of this invention to provide a set of algorithms which greatly simplify the compressed data manipulations required to bring about the corresponding image manipulations in the spatial domain.

[0008] It is still another object of this invention to provide a set of algorithms to manipulate image data in the compressed domain that includes a manipulation step which takes one set of compressed data and produces another set of compressed data, such that, if the process was reversed, the original image could be produced without any loss of quality.

[0009] The present invention provides a technique for performing various dihedral symmetry operations on a spatial domain representation of a distal image by manipulating a linear transform domain representation of the digital image. When a digital image is in the form of a compressed bitstream, such as a JPEG file, the technique involves entropy decoding the compressed bitstream to generate linear-transform-based data blocks which define the linear transform domain representation of the digital image. The linear-transform-based data blocks are reordered for the corresponding geometric transformation, a particular linear transform domain operation is applied to the data elements within each block, and the blocks are reassembled. When these blocks of data are decompressed into spatial domain, the resulting

digital image will be flipped or rotated with respect to the original image. With respect to the original image, the resulting image may be flipped over either diagonal (main or cross-), flipped over either of its middle axes (vertical or horizontal), or rotated 90°, 180° or 270°. The technique may be applied in an imaging device, such as a digital still-image camera, or in a computer system. In either case, the technique may be implemented using hardware or software.

[0010] Other objects and attainments together with a fuller understanding of the invention will become apparent and appreciated by referring to the following description and claims taken in conjunction with the accompanying drawings.

[0011] In the drawings wherein like reference symbols refer to like parts:

[0012] Fig. 1 is a block diagram depicting the JPEG compression and decompression processing flow.

[0013] Fig. 2 is a block diagram showing the spatial-domain based image processing for a JPEG data set.

[0014] Fig. 3 is a block diagram showing the compressed-domain based image processing for a JPEG data set, in accordance with the invention.

[0015] Fig. 4 is a diagram showing a block rotate-by-90° using diagonal-flip (F_d) and a column-flip (F_y) in accordance with the invention.

[0016] Fig. 5 is a block diagram of a digital still-image camera (DSC) which may be used in connection with the invention.

[0017] Fig. 6 is a block diagram that illustrates the interrelationship between various components that may be used in capturing and viewing digital images, as well as processing such images in accordance with the invention.

[0018] Since the 8 x 8 discrete cosine transform (DCT) is chosen as the basis function in still-image compression standards such as JPEG (see, G. K. Wallace, "The JPEG Still Picture Compression Standard," *Communications of the ACM,* vol. 34, no. 4, Apr. 1991), and since JPEG is the compression method widely used in many consumer digital still-cameras in the market, the following description will focus on some of the compressed-domain processing methods that are appropriate for digital still-camera applications and that operate on the DCT domain representations of the camera's JPEG compressed bitstream. Although 8 x 8 is the most common DCT block size, these techniques may be extended to other DCT block sizes. Furthermore, the techniques are extensible to any other linear transform based basis functions including discrete sine transform, discrete hadamard transform and also the wavelet transforms.

[0019] We begin by briefly describing the JPEG compression and decompression processes and the basic idea of incorporating image processing functions within the JPEG pipeline. JPEG uses the DCT to transform still-image data from its spatial or pixel domain representation to its compressed or frequency domain representation in which the data can be more efficiently coded. The image manipulation methods developed herein are designed to take advantage of the properties of the DCT.

[0020] The JPEG compression and decompression process, illustrated schematically in Fig. 1, operates on a block-by-block basis, where each block size is 8 x 8. As shown schematically in Fig. 1, the uncompressed still-image 11 is decomposed into 8 x 8 blocks of pixels by a raster-to-block converter 12. These blocks are then transformed by the forward 8 x 8 DCT 13 to produce a corresponding set of 8 x 8 DCT blocks. The forward 8 x 8 DCT $F(u,v)$ of a spatial domain 8 x 8 block of samples $f(i, j)$ is computed as:

$$F(u,v) = \frac{\alpha_u \alpha_v}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} C_{i,u} C_{j,v} f(i,j), \tag{1}$$

where,

$$C_{x,y} = \cos \frac{(2x+1)y\pi}{16}, \tag{2}$$

and,

$$\alpha_x = \begin{cases} 1, & x \neq 0 \\ 1/\sqrt{2}, & \text{otherwise.} \end{cases} \tag{3}$$

[0021] After output from the forward 8 x 8 DCT 13, each of the 64 DCT coefficients is uniformly quantized in a forward quantizer 14 in conjunction with a 64-element quantization table Q, which can be derived empirically to discard information which is not visually significant. In this compression process, the only loss incurred during the compression

comes from the quantization of F(u,v) to $F_Q(u,v)$ = RoundOff

$$\left(\frac{F(u,v)}{Q(u,v)}\right),$$

where Q is the 8 x 8 quantization table.

[0022] After quantization, the DCT data in each block is ordered into a "zigzag" sequence which facilitates entropy coding by placing low frequency coefficients (which are more likely to be non-zero) before the high frequency coefficients (which are more likely to be zero). The data is then Huffman coded in a Huffman encoder 15 to further compact the data and to generate a JPEG compressed bitstream.

[0023] The image may be reconstructed from the compressed bitstream using a symmetrical reverse process. The JPEG decompression process begins by decoding the compressed bitstream in a Huffman decoder 16 to regenerate the 8 x 8 blocks of DCT coefficients. The coefficients are reordered using an inverse zigzaging procedure and the blocks are then fed through an inverse quantizer 17. In the next step, the 8 x 8 inverse discrete cosine transform (IDCT) 18 operates on the 8 x 8 blocks of DCT coefficients to generate a stream of 8 x 8 blocks of pixels. A block-to-raster converter 19 converts these blocks into the decompressed still-image 21. In the decompression process, the IDCT converts the coefficients $F(u,v)$ back to the pixels $f(i, j)$, exactly:

$$f(i,j) = \sum_{u=0}^{7}\sum_{v=0}^{7}\frac{\alpha_u\alpha_v}{4}C_{i,u}C_{j,v}F(u,v). \qquad (4)$$

The decompression process will actually work with the quantized coefficients, F_Q, and obtain only an approximation f_Q of f.

$$f_Q(i,j) = \sum_{u=0}^{7}\sum_{v=0}^{7}\frac{\alpha_u\alpha_v}{4}C_{i,u}C_{j,v}F_Q(u,v)Q(u,v). \qquad (5)$$

[0024] If an image processing function such as rotation or scaling has to be performed on an image that is available only as a JPEG compressed bitstream, this processing could be performed as shown in Fig. 2. First, the JPEG compressed bitstream is decompressed in block 31 back into its spatial domain representation. Spatial domain image processing is then performed in block 32. Afterward, the processed pixel data is recompressed in block 33 to generate a new JPEG compressed bitstream. We refer to this scheme as a *spatial-domain approach* since the processing is directly applied to the decompressed spatial domain data (pixels).

[0025] The *spatial-domain approach* has certain advantages. One such advantage is that image processing in spatial domain is a well-understood problem and solutions are widely available for many typical image processing functions. Another advantage is that the processing function is independent of the underlying compression scheme used to represent the data.

[0026] This *approach* has certain disadvantages as well. The data has to be fully decompressed prior to applying the image processing function. Moreover, the processed data may have to undergo the compression process again. Since JPEG is a lossy compression method, decompression-recompression may result in a loss of image quality. For the portrait to landscape conversion application, the quality of an image would incrementally deteriorate every time its orientation is changed. Another disadvantage is that the complexity of the decompression and compression process is quite high. If, for instance, in Fig. 2, the image processing task is, say, clockwise rotation by 90°, then the operations count for rotating each 8 x 8 block of input data is as listed in Table 1. In order to get a rough estimate of the operations count, we have assumed that multiplies, adds and data accesses can each be done at one operation per data item.

Table 1:

Operations count for spatial-domain based rotate-by-90° operation when the input and output data is in 8 x 8 DCT form. [*] Y. Arai and T. Agui and M. Nakajima, "A Fast DCT-SQ Scheme for Images," *Trans. of the IEICE*, E 71(11): 1095, Nov. 1988.		
Task	Operations Count	Comments
Huffman Decode	81	assume 15% of IDCT'S complexity
Inverse Quantization	64	assume 1multiply requires 1 operation
8 x 8 IDCT	544	80 multiplies, 464 additions as per [*]
Rotate by 90°	128	simple matrix transposition (1 read, 1 write per pixel)
8 x 8 DCT	544	80 multiplies, 464 additions as per [*]
Forward Quantization	64	assume 1 multiply requires 1 operation
Huffman Encode	54	assume 10% of IDCT'S complexity
TOTAL	1415	

[0027] The present invention proposes an alternative to spatial-domain based image processing, namely, compressed domain based image processing. The latter is well suited for data that is already available in compressed form, such as a JPEG bitstream. The basic processing flow for a compressed-domain based image processing scheme is as depicted in Fig. 3. For JPEG data, compressed-domain based image processing usually implies DCT-domain processing which is represented by block 42 in Fig. 3.

[0028] As shown in Fig. 3, DCT-domain image processing 42 is preceeded by entropy decoding of the JPEG compressed bitstream in block 41 and followed by entropy coding in block 43. As also shown in Fig. 3, the DCT-domain image processing block 42 is further divided into processing blocks labelled 44-49. As previously noted, decompressing the bitstream yields 8 x 8 blocks of DCT coefficients 44. The coefficients are reordered using a de-zigzag procedure 45, after which the blocks of DCT coefficients are dequantized 46. In accordance with the invention, the dequantized blocks of DCT coefficients undergo the block-oriented processing 47. Following the processing, the DCT-data based blocks are quantized 48 and reordered into the zigzag sequence 49. However, depending on the image processing task, not all of the blocks 44-49 need to be performed. For instance, for the D_4 operations (which can be expressed as sequences of flips about the diagonal and the Y-axis), the de-zigzag 45, de-quantize 46, zigzag 49, and quantize 48 steps can be eliminated. The specifics of compressed domain based D_4 operations will be discussed in the next section.

[0029] In general, compressed-domain processing has the following advantages. First, image quality could be preserved since in many instances, dequantize-quantize steps can be avoided. Second, the complexity can be much lower than the spatial-domain counterpart depicted in Fig. 2 since full JPEG decompression and compression tasks are avoided. Specifically for the case of rotation by 90°, a simple implementation of the compressed-domain based processing counterpart of Table 1 has an operations count as shown in Table 2. Note that the total operations count using the compressed-domain approach is nearly five times lower than the spatial-domain based approach of Table 1. (Details of the DCT-domain based rotation method are discussed below.) Another adavantage is that, in typical digital imagery, due to high correlation among pixels, the DCT-domain representation tends to be quite sparse (e.g., in a 8 x 8 DCT block, usually there are around 7-16 nonzero values). This data sparseness property can be exploited by the DCT-domain processing approach to further reduce the overall complexity; this property is not available in a spatial-domain representation. (Special techniques to exploit this property for the D_4 operations are described below.)

Table 2:

Operations count for compressed-domain based rotate-by-90° operation when the input and output data is in 8 x 8 DCT form.		
Task	Operations Count	Comments
Huffman Decode	81	assume 15% of IDCT's complexity
Inverse Quantization	-	not needed
8 x 8 IDCT	-	not needed
Rotate by 90°	32	need a sign-change every odd column
	112	and row-column data interchange
8 x 8 DCT	-	not needed
Forward Quantization	-	not needed
Huffman Encode	54	assume 10% of IDCT's complexity
TOTAL	279	

[0030] Note that, in general, it may not be possible to derive a compressed-domain based processing equivalent of a spatial-domain based image processing function. DCT is a linear transform and hence, compressed-domain based processing can probably be accomplished for linear image processing functions. Nonlinear image processing functions such as median-filtering, warping/morphing are not amenable to compressed-domain based approaches such as the one depicted in Fig. 3.

DIHEDRAL SYMMETRY OPERATIONS ON JPEG IMAGES

[0031] In this section, we develop the basic equations governing simple geometric transformation of JPEG compressed data. The operations defined by compositions of flips about the diagonal axes, the Y-axis (i.e., the middle-vertical axis) and the X-axis (i.e., the middle-horizontal axis) form the group of dihedral symmetry of the square, referred to as D_4. These operations are listed and described in Table 3.

Table 3:

The group D_4 of dihedral symmetry of a square.		
Operation	Description	Generation by F_d and F_y
I	Identity	F_dF_d
F_d	Diagonal-flip (flip over main diagonal)	F_d
F_y	Column-flip (flip over Y-axis)	F_y
F_{cd}	Cross-diagonal-flip (flip over cross-diagonal)	$F_yF_dF_y$
F_x	Row-flip (flip over X-axis)	$F_dF_yF_d$
R_{90}	Rotate 90° clockwise	F_yF_d
R_{180}	Rotate 180°	$F_yF_dF_yF_d$
R_{-90}	Rotate 90° counterclockwise	F_dF_y

[0032] Note that the operations F_d and F_y can be composed to generate the whole group. (Notational convention: the composition o_1o_2 of operations o_1 and o_2 is the operation resulting from first applying o_2 and then o_1.) For example, a simple clockwise rotate-by-90° (R_{90}) can be achieved by applying a diagonal-flip followed by a column-flip as depicted in Fig. 4. Thus, if we can derive the compressed-domain based counterparts of the two operations F_d and F_y, all the rest can be derived from them.

[0033] Let *f* be the 8 x 8 pixel block, and *F* be the corresponding 8 x 8 DCT block (DCT(*f*) = *F* and IDCT(*F*) = *f*). For any of the eight D_4 operations, o, it is easy to express the relationship between *f* and o*f*. The goal is to derive the relationship between *F* and *DCT(of)* (denoted by o*F*).

[0034] Consider the operation F_y. In spatial domain, the column-flip output $F_y f(i,j)$ can be expressed as:

$$F_y f(i,j) = f(i,7-j). \tag{6}$$

[0035] From Eq. 1, $F_y F(u,v)$ is:

$$\mathbf{F}_Y F(u,v) = \frac{\alpha_u \alpha_v}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} C_{i,u} C_{j,v} \, \mathbf{F}_Y f(i,j) \tag{7}$$

$$= \frac{\alpha_u \alpha_v}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} C_{i,u} C_{j,v} f(i,7-j) \tag{8}$$

$$= \frac{\alpha_u \alpha_v}{4} \sum_{i=0}^{7} \sum_{k=0}^{7} C_{i,u} C_{7-k,v} f(i,k), \tag{9}$$

where, $k = 7\text{-}j$.

[0036] Using Eq. 2, $C_{7-k,v}$ can be expressed as:

$$C_{7-k,v} = \cos\left(\frac{(2(7-k)+1)v\pi}{16}\right) \tag{10}$$

$$= \cos\left(v\pi - \frac{(2k+1)v\pi}{16}\right) \tag{11}$$

$$= \cos(v\pi)\cos\left(\frac{(2k+1)v\pi}{16}\right) \tag{12}$$

$$= (-1)^v \cos\left(\frac{(2k+1)v\pi}{16}\right) \tag{13}$$

$$= (-1)^v C_{k,v}. \tag{14}$$

[0037] Using Eq. 14 and the DCT definition in Eq. 1, $F_y F(u,v)$ in Eq. 9 can be rewritten as:

$$\mathbf{F}_Y F(u,v) = \frac{\alpha_u \alpha_v}{4} (-1)^v \sum_{i=0}^{7} \sum_{k=0}^{7} C_{i,u} C_{k,v} \, f(i,k) \tag{15}$$

$$= (-1)^{v} F(u,v). \qquad (16)$$

[0038] In spatial-domain, diagonal-flip of an input block $f(i,j)$ is $F_d\ f(i,j) = f(j,i)$. It can be easily seen that the DCT-domain equivalent of this is:

$$F_d F(u,v) = F(v,u). \qquad (17)$$

[0039] Using Eq. 15, Eq. 17, and the relationships given in the third column of Table 3, we can derive the compressed domain counterparts for all the D_4 operations. These are listed in Table 4.

Table 4:

DCT-domain operations for geometric transformations in D_4.		
Spatial Domain	DCT-domain	Input block processing
$I f(i,j) = f(i,j)$	$I F(u,v) = F(u,v)$	None
$F_d\ f(i,j) = f(j,i)$	$F_d\ F(u,v) = F(v,u)$	Transpose
$F_y f(i,j) = f(i,7-j)$	$F_y F(u,v) = (-1)^v\ F(u,v)$	Sign-reverse odd columns
$F_{cd} f(i,j) = f(7-j,7-i)$	$F_{cd} F(u,v) = F_y F_d F_y F(u,v)$ $= (-1)^v\ F_d F_y F(u,v)$ $= (-1)^v\ F_y F(v,u)$ $=(-1)^v\ (-1)^u\ F(v,u)$	Transpose and sign-reverse every other element
$F_x f(i,j) = f(7-i,j)$	$F_x F(u,v) = F_d F_y F_d\ F(u,\ v)$ $= F_y F_d F(v,u)$ $= (-1)^u\ F_d F(v,u)$ $= (-1^u)\ F(u,v)$	Sign-reverse odd rows
$R_{90}\ f(i,j) = f(7-j,i)$	$R_{90}\ F(u,v) = F_y F_d F(u,v)$ $= (-1)^v\ F_d F(u,v)$ $= (-1)^v\ F(v,u)$	Transpose and sign-reverse odd columns
$R_{180}\ f(i,j) = f(7-i,7-j)$	$R_{180}\ F(u,v) =$ $F_y F_d F_y F_d\ F(u,v)$ $= (-1)^v\ F_d F_y F_d\ F(u,v)$ $= (-1)^v\ F_y F_d F(v,u)$ $= (-1)^v\ (-1)^u\ F_d F(v,u)$ $=(-1)^v\ (-1) u\ F(u,v)$	Sign-reverse every other element
$R_{90}\ f(i,j) = f(j,7-i)$	$R_{-90}\ F(u,v) = F_d F_y F(u,v)$ $= F_y F(v,u)$ $= (-1)^u\ F(v,u)$	Transpose and sign-reverse odd rows

[0040] The above relationships were derived without taking quantization into account. By simply using the dequantized coefficients, $F_Q(u,v)Q(u,v)$, in place of $F(u,v)$, it can be easily seen that the quantized coefficients of a block produced by a D_4 operation can be directly obtained by applying the corresponding input block processing (column 3 of Table 4) on the quantized coefficients of the original block. The quantization table remains the same (it is transposed for F_d, F_{cd}, R_{90}, and R_{90}). This DCT-domain approach avoids the IDCT, the DCT, as well as dequantization and quantization. It will be shown below that the de-zigzag and zigzag steps can also be avoided.

[0041] In order to perform a D_4 operation on an W x H JPEG image consisting of many 8 x 8 blocks (W and H in any JPEG image are multiples of 8, obtained by padding the original image, if necessary), the compressed-domain based procedure is as follows: (a) reorder the 8 x 8 DCT blocks for the corresponding geometric transformation, and (b) apply the DCT-domain operation as per Table 4 to elements within each 8 x 8 DCT block. Note that (a) and (b) are *lossless operations* in that the quantized DCT coefficients $F_Q(u,v)$ are not manipulated beyond a sign-change; by avoiding the process of dequantization and requantization, no quality loss is incurred regardless of the number of times one or more geometric transformations of Table 4 are applied to the camera JPEG file.

[0042] For simplicity, we first describe the implementation techniques using grayscale images; the simple modifica-

tions needed for color images are outlined subsequently. We assume that the width and height of the image are multiples of 8. Moreover, for color images, we will assume that subsampling does not necessitate block padding. All the operations described here can also be applied when these constraints are not satisfied, by first padding the original image with some extra rows/columns.

[0043] Consider a $W \times H$ grayscale image, I, available as JPEG data. By applying entropy-decoding to the JPEG data, and undoing the differential coding for the DC terms, we can obtain the quantized DCT coefficients for each block. Let F_k denote the 8 x 8 block (numbered k in raster-order) of quantized DCT coefficients for the image ($0 \leq k < WH/64$).

[0044] Let I^o denote the result of applying operation o on the image (where o is one of the D_4 operations from Table 3). From the preceding discussion, it is apparent that the DCT coefficient blocks of I^o will essentially be the same as those in I, with possible reordering, transposition, and sign changes, and the quantization table will also be the same, with possible transposition. In general, the block F_k^o of quantized coefficients in I^o, can be expressed as:

$$F_k^o = oF\,p^o_{(k)}$$

where p^o is a permutation of the blocks. Consider, as an example, clockwise rotation by 90° (o = R_{90}). For $k = i_b\,(H/8) + j_b$ (that is the block in row i_b and column j_b of blocks, in the rotated image I^o), we have,

$$p^o(k) = (H - j_b - 1)\,(W/8) + i_b.$$

[0045] In order to create the JPEG image I^o, we need to calculate the blocks F_k^o in the order, $k = 0, 1, 2, \ldots$ This would require accessing the blocks of I in the order $p^o(0), p^o(1), p^o(2), \ldots$, which is (in general) different from the raster-order in which these blocks are stored in the JPEG image I. Since the compressed blocks can be of arbitrarily different sizes, extraction of a particular block from the JPEG bitstream would require parsing of all the prior blocks, which is too expensive. Moreover, the value of the quantized DC coefficient in any block can only be extracted after all the previous DC values are decoded, because of differential coding. We avoid these problems by using a two-pass approach. In the first pass, the JPEG image I is parsed to extract the bit-offset for each block and the DC value for each block. In the second pass, when computing F_k^o for the image I^o, the block $F_{p^o(k)}$ is readily accessed by looking up the bit-offset for block number $p^o(k)$ and seeking to that position in the JPEG bitstream for I. This algorithm is summarized in the following pseudo-code.

procedure OperateJPEG
input: JPEG image I, operation o
output: JPEG image I^o
/* first pass: gather offsets and DC values */
parse off the header from I
for $k = 0$ to $WH/64 - 1$

1. store the offset into I as the bit-offset for block k
2. parse I to move past another block while extracting the DC term
3. undo the differential coding and store the DC value for block k

/* second pass: compute and store I° */
compute and store header for I^o, transposing the quantization table of I if required by o for $k = 0$ to $WH/64 - 1$

1. use the offset and DC tables to seek into I and extract the quantized coefficients of block number $p^o(k)$
2. transpose and/or apply sign changes to the coefficients, as required by o (Table 4, column 3)
3. append the entropy-coded coefficients to the JPEG bitstream for I^o

EFFICIENT TRANSPOSITION AND SIGN-CHANGING OF BLOCKS

[0046] The heart of the "OperateJPEG" algorithm lies in the second pass, when the original block $F_{p^o(k)}$ is transformed via o to get the block F_k^o for the image I^o. Transposing and applying sign-changes to an 8 x 8 block would require accessing each element of the block once, if implemented in a straight-forward manner. That is, the complexity will be a multiple of 64. The quantized coefficient blocks in typical JPEG image have a very small number (typically less than 16) of non-zero coefficients. We now present an algorithm to do the transposition and sign-changes with a complexity proportional to the number of non-zero coefficients in the block. This results in a substantial decrease in the overall complexity, making it linear in the total size of the *compressed* image, rather than in the total size of the uncompressed

image.

[0047] In the Huffman coding mode of JPEG a block of quantized coefficients is coded by scanning it in zigzag order to group together long runs of zeros. The zigzag ordering is specified in Table 5. When coding a block, first the difference between the quantized DC value of the current block and the previous block is coded, using a Huffman table specific for these differentials.

Table 5:

The zigzag ordering, *ZZ*(*u*,*v*), used by JPEG.								
	0 v→ 7							
0	0	1	5	6	14	15	27	28
	2	4	7	13	16	26	29	42
	3	8	12	17	25	30	41	43
	9	11	18	24	31	40	44	53
u ↓	10	19	23	32	39	45	52	54
	20	22	33	38	46	51	55	60
	21	34	37	47	50	56	59	61
7	35	36	48	49	57	58	62	63

[0048] Next, the quantized AC coefficients are scanned in zigzag order and coded as a sequence of symbol trios of the form (R, S, V), where R is the number of consecutive zeros in the zigzag ordering, and the next non-zero coefficient has the value x such that:

$$S = \lceil \log_2 (|x| + 1) \rceil_p$$

$$V = S \text{ least significant bits of } \begin{cases} x & \text{when } x > 0 \\ x-1 & \text{when } x < 0. \end{cases}$$

[0049] The bits for V are extracted using the standard 2's complement binary representation of x or x-1. In the JPEG bitstream, each (R, S, V) is coded using first a Huffman table to encode (R, S) and then S extra bits to encode V. There are some special coding situations such as very long runs of zeros and end-of-block, handled by special codes. But for our purposes, it should be clear from this description that the JPEG data can be easily parsed to capture each block in a data structure of the form:

```
struct JpegBlock {
    int D; /* the quantized DC coefficient */
    int N; /* number of non-zero AC coefficients */
    struct {
        int Z; /* the zigzag index */
```

```
            int S;
            int V;
        } A[63];
};
```

[0050] In this data structure, N is the number of non-zero AC coefficients in the block. Only the first N entries of the array A are meaningful. The element $A[k]$ gives the zig-zag location (Z), and the S and V values for the k^{th} non-zero AC coefficient, $0 \leq k < N$. Reading a block from a JPEG bitstream into this structure, and writing this structure as JPEG data, are both straightforward, and the details are omitted here.

[0051] The operation o may require transposition and/or sign changes. We now describe a novel algorithm for implementing o, using the above data structure, such that only N steps are needed (rather than 64). This algorithm avoids de-zigzagging and zigzagging of the coefficients, as it uses the zigzag-ordered block representation directly.

[0052] Given an input "JpegBlock" structure B (corresponding to a block F), we would like to find the structure B^o, corresponding to oF. For changing the sign of a non-zero quantized AC coefficient, x, with the corresponding S and V values being S_x and V_x, it is sufficient simply to take the bitwise complement of V_x. That is,

$$S_{-x} = S_x,\ V_{-x} = -V_x.$$

[0053] Let $s^o[64]$ be a precomputed array of boolean flags such that $s^o[Z]$ is TRUE if and only if o requires the sign of the Z^{th} zigzag coefficient to be changed. If only sign-changes are needed (i.e., for the operations F_x, F_y, R_{180}), then, B can be converted into B^o by copying each entry, while flipping the bits for those $A[k].V$ for which $s^o[A/k].Z]$ is TRUE.

[0054] Transposing a "JpegBlock" structure in N steps uses the following key observation: the array A remains the same, except that elements on any given cross-diagonal are reversed in order. To illustrate this, let t[Z] denote the transposed zigzag index for the original zigzag index Z. That is,

$$t[1]= 2,\ t[2] = 1,\ t[3] = 5,\ t[4] = 4, \ldots$$

Consider a situation when the zigzag coefficients numbered:

1, 3, 6, 7, 10, 11, 14

are the only non-zero coefficients. Then, the zigzag ordering after transposition will be:

<u>*t[1], t[3], t[7], t[6], t[14], t[11], t[10].*</u>

[0055] Thus, each group of coefficients lying on the same cross-diagonal needs to be reversed. This can be done efficiently by scanning the array $B.A[\ldots]$ of non-zero coefficients, pausing at each coefficient where the cross-diagonal changes to copy the previous cross-diagonal from $B.A[\ldots]$ to $B^o.A[\ldots]$ in reverse order.

[0056] The following piece of pseudo-code summarizes the algorithm. The arrays t and s^o (defined previously) are precomputed. In addition, let $d[64]$ be another precomputed array, giving the "cross-diagonal number" for each zigzag index. That is, if the zig-zag index Z corresponds to row number u and column number v, then, $d[Z] = u + v$.

procedure TransposeAndSignChange

input: JpegBlock B, arrays t, s^o, d

output: JpegBlock B^o

```
Bo.D = B.D /* note that so[0] is always FALSE */
Bo.N = B.N
if (B.N == 0) then return
dcur = d[B.A[0].Z] /* the starting cross-diagonal number */
startpos = 0 /* where cross-diagonal number dcur begins in B.A[...] */
for k = 0 to B.N
    /* find the new cross-diagonal number */
    if (k < B.N) then dnew = d[B.A[k].Z]
    else dnew = ∞ /* any value > 14, just to ensure that dnew > dcur */
    if (dnew > dcur) then
        /* copy B.A[startpos ...(k-1)] into Bo.A[...] in reverse order */
        for l = 0 to k - startpos - 1
            i = k - 1 - l /* index in B.A[...] */
            j = startpos + l /* index in Bo.A[...] */
            Bo.A[j].Z = t[B.A[i].Z]

            Bo.A[j].S = B.A[i].S
            Bo.A[j].V = B.A[i].V
            if (so[Bo.A[j].Z]) then
                Bo.A[j].V = ¬Bo.A[j].V /* flip the bits */
        /* update dcur and startpos */
        dcur = dnew
        startpos = k
```

Color images

[0057] In general, a JPEG image I consists of more than one color plane, with some planes subsampled. Let the number of color planes be denoted by P. Associated with each plane, p ($1 \le p \le P$), there is a horizontal sampling factor, w_p, and a vertical sampling factor, h_p. The width and height of plane number p are given by $W\, w_p / w_{max}$ and $H\, h_p / h_{max}$, respectively. Here, w_{max} is the maximum value of w_p, and h_{max} is the maximum value of h_p, over all planes ($1 \le p \le P$).

[0058] The JPEG data can have a further layer of structuring, consisting of scans. A scan consists of one or more color planes, with possibly a limited bit-precision of coefficient values. The "OperateJPEG" algorithm can be extended simply by applying it to each scan in succession.

[0059] Within each scan, the data from the blocks of all the color planes in the scan is organized into units known as minimum coded units (MCUs). Each MCU consists of a fixed number of blocks in a fixed order, determined by all the w_p and h_p. For the operations we are considering, the image I^o consists of exactly the same scans and MCU groupings as I, as long as we do not change the sampling factors (h_p and w_p *do* need to be swapped for the operations requiring transposition). The order of the MCUs and the order of blocks within each MCU might change, but each MCU in I^o can be obtained from exactly one MCU in I. We exploit this fact to also save some memory by building the bit-offset and DC tables at MCU level, rather than block level.

[0060] Thus, for every scan in I, "OperateJPEG" works in two passes. In the first pass, for each MCU in the scan,

the bit-offset is recorded, and the DC value of the first block in that MCU from each color plane is recorded. In the second pass, when a particular MCU for the scan of I^p is to be computed, the corresponding MCU of I is extracted and the "JpegBlock" structure for each constituent block is filled out. These blocks are then transposed/sign-changed/reordered (as required by the operation) to form the MCU for I^p which is written out as JPEG data.

COMPRESSED DOMAIN BASED IMAGE ROTATIONS - DSC APPLICATION

[0061] The compressed-domain processing techniques of the present invention may be employed in connection with various digital devices including a digital still-image camera (DSC), a block diagram of which is illustrated in Fig. 5. Operating under microprocessor control, the DSC 60 has a charge-coupled device (CCD) image sensor that captures an image and converts it to an analog electrical signal in block 61. The analog signal is then processed and digitized in block 62, after which the digital image is temporarily stored in a frame buffer 63 while it undergoes digital processing in block 64. The digital image processing block 64 performs several functions including compression and decompression and may also perform the compressed-domain based processing techniques of the present invention. Under user control 65, the processing block 64 interfaces with in-camera image storage 66 where decompressed image data may be stored. The storage block 66 may comprise compact magnetic or solid-state storage media, either removable or fixed within the DSC 60, and may include removable, large-capacity PCMCIA-format hard disk cards or flash memory cards.

[0062] The DSC 60 includes analog and digital outputs, 67 and 68 respectively, through which image data may be transmitted within the DSC or to external devices. Uncompressed image data may be transmitted, via the analog outputs 67, to an LCD screen 69 within the DSC 60, or to external devices such as a VCR or TV monitor. Image data, whether compressed or uncompressed, may also be transmitted through the digital outputs 68 to a digital device such as a computer system where the image could be displayed.

[0063] The ability to perform the D_4 operations such as rotation by 90°,180°,270° or mirror-flip by directly manipulating the DCT-domain representations is quite useful in the DSC 60. When a camera is held vertically to capture a tall object, the resulting image appears as an image that has undergone a 90° rotation with respect to the same picture taken with the camera held horizontally. With typical film cameras this is not a problem, since when holding the prints, one simply rotates the picture by 90° to view the correct image. With DSCs 60 pictures taken in the manner described above can be problematic since often these images are directly printed or viewed on a TV monitor or incorporated in a computer document. By incorporating compressed-domain based image processing, during readout of the DSC image file, the DSC 60 can produce the corrected view of the image by undoing the rotation as per the methods described herein. In a DSC system, the rotation function can be within the DSC 60 or within the software driver running on a computer which is connected to the DSC 60. For a DSC 60 having limited computing capabilities and limited memory, a compressed-domain based approach might be the only efficient way in which to perform the geometric transformation. The alternative approach of decompressing the data and then performing the manipulation is memory intensive and the slower CPU speed in the DSC 60 can lead to significant performance penalties. As DSC systems evolve, additional features such as image enhancement and watermarking can be easily incorporated within the compressed domain processing framework without having to go back-and-forth between a compressed domain representation and spatial-domain representation.

[0064] As noted above, the DSC 60 shown in Fig. 5 can also be used in connection with a computer system and other components in capturing, processing and viewing digital images. Fig. 6 is a block diagram that illustrates the interrelationship between the DSC 60, a computer system and various other components. The computer system, identified generally by reference numeral 100, may be of any suitable type such as a main frame or personal computer.

[0065] Computer system 100 comprises a central processing unit (CPU) 101 which may be a conventional microprocessor, a random access memory (RAM) 102 for temporary storage of information, and a read only memory (ROM) 103 for permanent storage of information. Each of these components is coupled to a bus 104. Operation of the computer system 100 is typically controlled and coordinated by operating system software. The operating system, which is embodied in the system memory and runs on CPU 101, coordinates the operation of computer system 100 by controlling allocation of system resources and performing a variety of tasks, such is processing, memory management, networking and I/O functions, among others.

[0066] Also coupled to bus 104 by a controller 105 is a diskette drive 106 into which a non-volatile mass storage device such as a diskette 107 may be inserted. Similarly, a controller 108 interfaces between bus 104 and a compact disc (CD) ROM drive 109 which is adapted to receive a CD ROM 110. A hard disk 111 is provided as part of a fixed disk drive 112 which is coupled to bus 104 by a disk controller 113.

[0067] Software for the compressed-domain based processing techniques may be stored on storage devices 107 and 110 and transferred to CPU 101 for execution. Alternatively, the software may be stored in RAM 102 or ROM 103. Similarly, image data processed or to be processed in accordance with the invention may be loaded into and extracted from computer system 100 using removable storage media devices such as the diskette 107 and CD ROM 110.

[0068] Image data may be input into computer system 100 in other ways as well. Film-based images 114 generated by a film camera 115 can be digitized by a scanner 116 for storage and processing by the computer 100. The DSC 60 can directly digitize images and transmit them to the computer 100, as previously explained. A keyboard 121 and mouse 122, which are coupled to bus 104 via a controller 123, facilitate the input of such data and otherwise provide a means for entering information into computer system 100.

[0069] Image data may also be transferred to and from computer 100 for remote locations. To this end, computer 100 may also include a communications adapter 124 which enables the computer 100 to communicate with networks 125, which may include local area networks (LANs), the internet or online services, via direct connections or via modem.

[0070] Digital images transmitted or stored in computer 100 may be viewed in a number of different ways. A printer 126 attached to computer 100 can produce color prints that vary in quality depending on the printer 126. Another option is to view the images on a display 127 associated with the computer 100. Yet another choice is to display the images on a television receiver using a VCR.

[0071] As the foregoing description demonstrates, the inventors herein have developed a simple compressed-domain based processing framework for JPEG compressed still imagery. It has been shown that simple geometric transformations such as mirror image, rotations by 90°, 180° and 270° can be easily performed in the DCT domain without any loss of image fidelity. Furthermore, it was also shown that even a simple transformation such as rotate-by-90° can achieve significant speedup when performed in DCT domain instead of the conventional spatial-domain based processing approach. The practical use of the compressed-domain based image transformations in a real-world setting, namely a digital still-camera has also been described. Since images captured by a digital camera are quite noisy to begin with, it is imperative that any image processing applied to these images should not increase the noise and the methods of the present invention are essentially *lossless operations*. Furthermore, these methods are well suited to the limited computing and memory capabilities found in consumer digital still cameras. The basic compressed domain framework can be used for other types of image transformations; specifically, the methodology described here can be extended to image enhancement, image filtering (see, B. Chitprasert and K. R. Rao, "Discrete Cosine Transform Filtering," *Signal Processing,* vol 19, pp. 233-245, 1990), resolution translation, etc.

[0072] While the invention has been described in conjunction with specific embodiments, it will be evident to those skilled in the art in light of the foregoing description that many further alternatives, modifications and variations are possible. For example, the block diagrams used to illustrate the compressed-domain based processing techniques of the present invention, show the performance of certain specified functions and relationships thereof. The boundaries of these functional blocks have been arbitrarily defined herein for the convenience of description. Alternate boundaries may be defined so long as the specified functions and relationships thereof are appropriately formed. Moreover, the psuedo-code used to illustrate the algorithms of the present invention does not depict syntax or any particular programming language. Rather, it provides the functional information one skilled in the art would require to fabricate circuits or to generate software to perform the processing required. Each of the functions depicted in the block diagrams may be implemented, for example, by software instructions, a functionally equivalent circuit such as a digital signal processor circuit, an application specific integrated circuit (ASIC) or combination thereof. The present invention, having been thus described, is intended to embrace all such alternatives, modifications, applications and variations as may fall within the scope of the appended claims.

Claims

1. A method for performing a dihedral symmetry operation on a spatial domain representation of a digital image by manipulating a linear transform domain representation of the digital image, said method comprising the steps of:

   extracting a plurality of linear-transform-based data blocks defining the linear transform domain representation of the digital image;
   reordering the plurality of linear-transform-based data blocks;
   applying a linear transform domain operation to at least one of the plurality of linear-transform-based data blocks; and
   reassembling the plurality of linear-transform-based data blocks; and
   wherein the spatial domain representation of the digital image undergoes a dihedral symmetry operation.

2. The method as recited in Claim 1, further comprising the steps of:

   parsing the compressed image data to extract an offset value and a DC value for each of the plurality of linear-transform-based data blocks;
   accessing each of the plurality of linear-transform-based data blocks by using the corresponding offset value;

and
extracting the elements of each the plurality of linear-transform-based data blocks.

3. The method as recited in Claim 1, wherein the dihedral symmetry operation comprises any one of the following operations namely: flipping the digital image over its main diagonal; flipping the digital image over its middle-vertical axis; flipping the digital image over its cross-diagonal axis; flipping the digital image over its middle-horizontal axis; rotating the digital image 90° clockwise; rotating the digital image 180°; rotating the digital image 90° counterclockwise.

4. The method as recited in Claim 1, wherein the dihedral symmetry operation comprises flipping the digital image over its main diagonal and wherein the linear transform domain operation is applied to each of the plurality of linear-transform-based data blocks to transpose the elements within each block.

5. The method as recited in Claim 1, wherein the dihedral symmetry operation comprises flipping the digital image over its middle-vertical axis and wherein the linear transform domain operation is applied to each of the plurality of linear-transform-based data blocks to sign-reverse the odd-column elements within each block.

6. The method as recited in Claim 1, wherein the dihedral symmetry operation comprises flipping the digital image over its cross-diagonal axis wherein the linear transform domain operation is applied to each of the plurality of linear-transform-based data blocks to transpose the elements within each block and then sign-reverse every other element within each block.

7. The method as recited in Claim 1, wherein the dihedral symmetry operation comprises flipping the digital image over its middle-horizontal axis and wherein the linear transform domain operation is applied to each of the plurality of linear-transform-based data blocks to sign-reverse the odd-row elements within each block.

8. The method as recited in Claim 1, wherein the dihedral symmetry operation comprises rotating the digital image 90° clockwise and wherein the linear transform domain operation is applied to each of the plurality of linear-transform-based data blocks to transpose the elements within each block and then sign-reverse the odd-column elements within each block.

9. The method as recited in Claim 1, wherein the dihedral symmetry operation comprises rotating the digital image 180° and wherein the linear transform domain operation is applied to each of the plurality of linear-transform-based data blocks to sign-reverse every other element within each block.

10. The method as recited in Claim 1, wherein the dihedral symmetry operation comprises rotating the digital image 90° counterclockwise and wherein the linear transform domain operation is applied to each of the plurality of linear-transform-based data blocks to transpose the elements within each block and then sign-reverse the odd-row elements within each block.

11. A digital camera, comprising:

a sensor for capturing light and converting the light into an analogue image signal;
an analogue-to-digital converter for converting the analogue image signal to a digital image in spatial domain; and
a digital image processor for compressing the digital image from a spatial domain representation of the digital image to a linear transform domain representation of the digital image defined by a plurality of linear-transform-based data blocks, wherein said digital image processor extracts the plurality of linear-transform-based data blocks, reorders the plurality of linear-transform-based data blocks, applies a linear transform domain operation to at least one of the plurality of linear-transform-based data blocks, and reassembles the plurality of linear-transform-based data blocks to perform a dihedral symmetry operation on the spatial domain representation of the digital image.

12. A computer system having a computer-readable program code embodied therein for causing the computer system to geometrically transform a digital image in spatial domain by performing the step of:

processing the digital image to obtain a linear transform domain representation of the digital image defined by a plurality of linear-transform-based data blocks; and

performing the steps of the method of any one of claims 1 to 10.

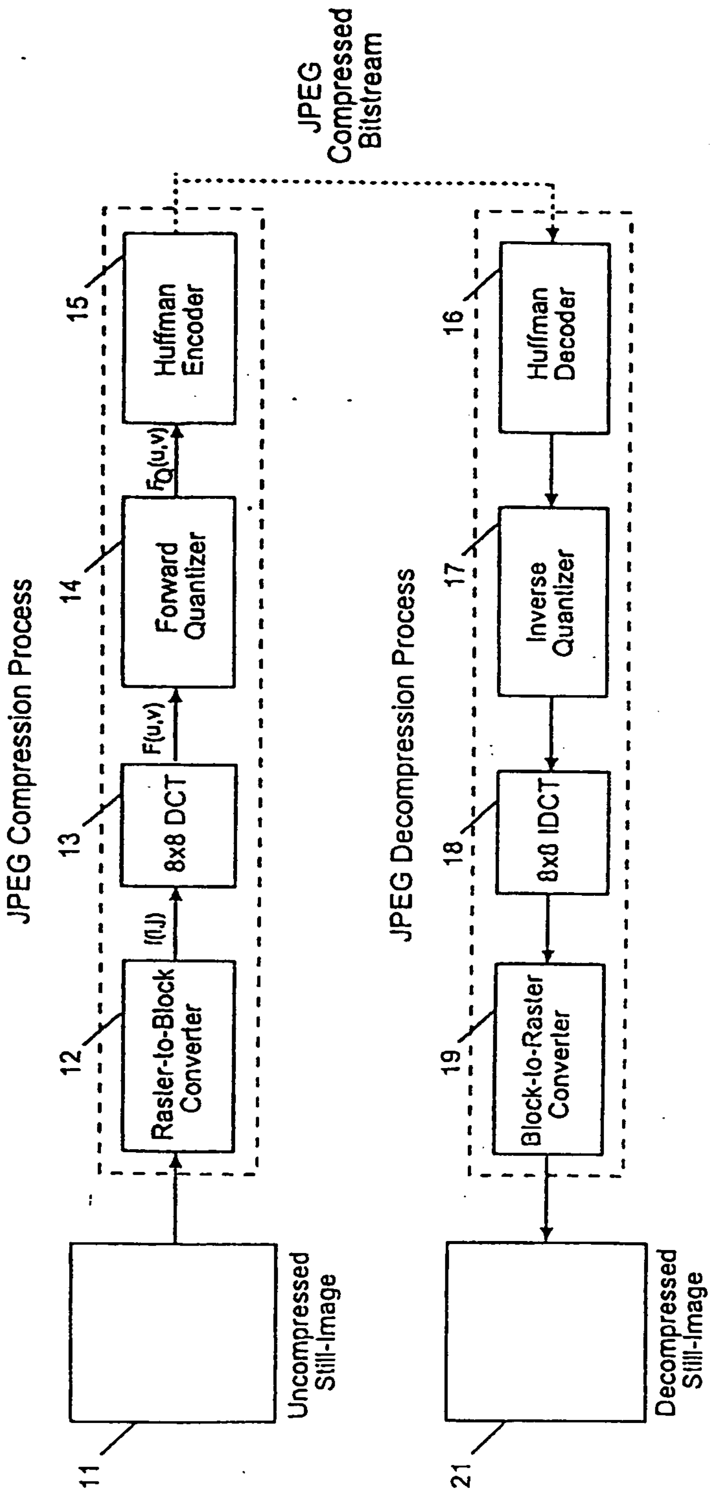
JPEG Compression Process
11
Uncompressed Still-Image
12
Raster-to-Block Converter
f(i,j)
13
8x8 DCT
F(u,v)
14
Forward Quantizer
$F_Q(u,v)$
15
Huffman Encoder
JPEG Compressed Bitstream
JPEG Decompression Process
16
Huffman Decoder
17
Inverse Quantizer
18
8x8 IDCT
19
Block-to-Raster Converter
21
Decompressed Still-Image

Fig. 1

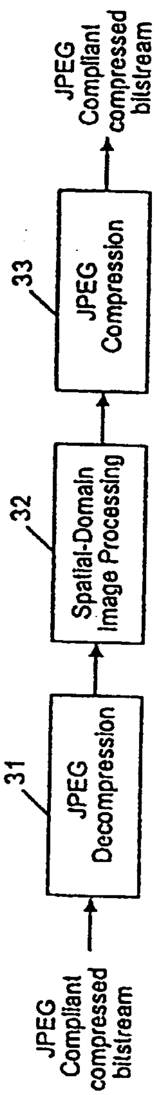
JPEG Compliant compressed bitstream
31
JPEG Decompression
32
Spatial-Domain Image Processing
33
JPEG Compression
JPEG Compliant compressed bitstream

Fig. 2

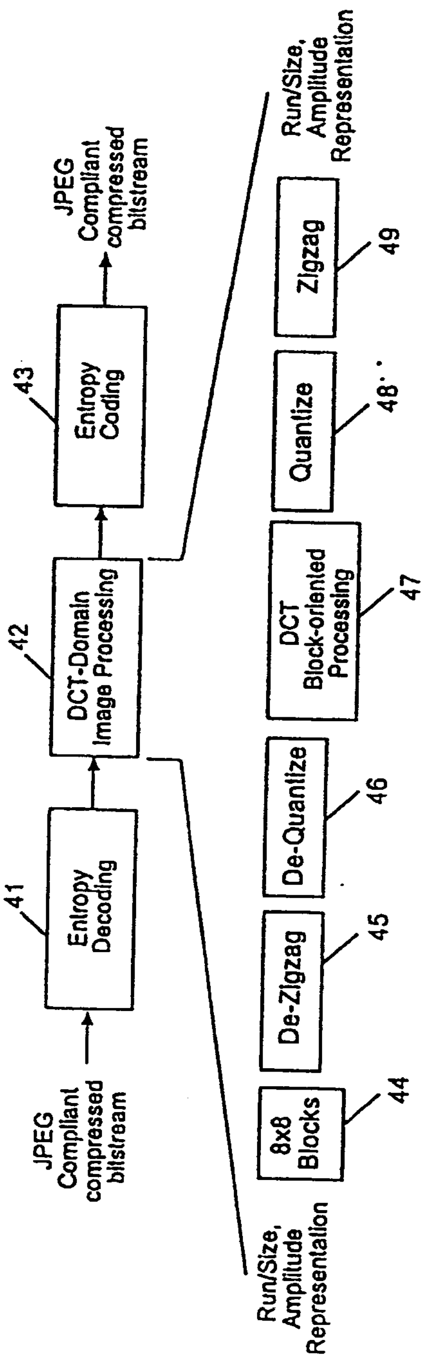
JPEG Compliant compressed bitstream
41
Entropy Decoding
42
DCT-Domain Image Processing
43
Entropy Coding
JPEG Compliant compressed bitstream
Run/Size, Amplitude Representation
8x8 Blocks
44
De-Zigzag
45
De-Quantize
46
DCT Block-oriented Processing
47
Quantize
48
Zigzag
49
Run/Size, Amplitude Representation

Fig. 3

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 982 931 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
01.03.2000 Bulletin 2000/09

(51) Int Cl.7: H04N 1/41

(21) Application number: 99401970.1

(22) Date of filing: 03.08.1999

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 26.08.1998 FR 9810734
02.12.1998 FR 9815212

(71) Applicant: CANON KABUSHIKI KAISHA
Tokyo (JP)

(72) Inventors:
- Berthelot, Bertrand
35200 Rennes (FR)
- Henry, Félix
35700 Rennes (FR)
- Majani, Eric
35000 Rennes (FR)

(74) Representative: Rinuy, Santarelli
14, avenue de la Grande Armée,
B.P. 237
75822 Paris Cédex 17 (FR)

(54) Geometric transcoding of a digital signal

(57) A method for the geometric transcoding of a compressed data file (F) containing a digital signal of dimension N coded by means of a coding method including at least one step of spectral breakdown into frequency sub-bands of the digital signal (S) includes the following steps:

- extracting (E9) symbols associated with the coefficients of the frequency sub-bands of the digital signal;
- applying (E10) a geometric transformation to said symbols;
- updating (E12, E13) N indicator or indicators representing a normal or reversed order of the symbols respectively in N direction or directions of the digital signal and
- reconstituting (E18) the digital signal coded by reversal of the extraction step (E9).

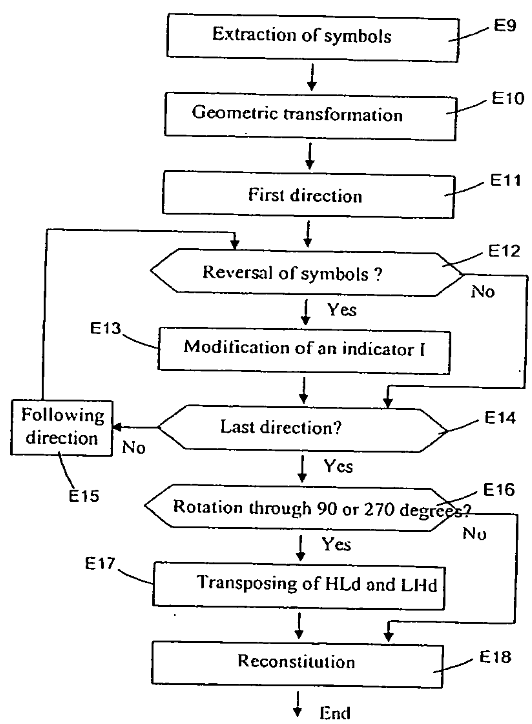

Figure 9

EP 0 982 931 A1

Description

[0001] The present invention concerns a method and a device for the geometric transcoding of a compressed digital signal. It also concerns a coding method and device on the one hand, and a decoding method and device on the other hand, associated with the geometric transcoding method and device.

[0002] The technical field of the invention is that of methods for manipulating compressed digital data.

[0003] Geometric transcoding is a transformation of one compressed digital data file into another compressed digital data file which, once decoded, represents a digital signal which has undergone a geometric transformation with respect to the initial coded digital signal in the initial compressed data file.

[0004] The geometric transformation of a mono- or multi-dimensional digital signal consists of changing the order of the samples of digital signal in one or more directions of the digital signal.

[0005] In practice, in the case for example of an image, axial or central symmetry can be produced, or a rotation of the image by manipulating the digital signal representing the image.

[0006] It is particularly advantageous to be able to carry out this manipulation and transformation of a coded digital signal without having to decompress and then once again compress the file containing the coded digital signal, in order notably to reduce the number of calculations and the memory space necessary for a decompression of the compressed digital data file.

[0007] Techniques are known for manipulating digital data in a compressed form, which have been coded by a coding method based on a discrete cosine transform (DCT). Such techniques are described for example in the article "A new family of algorithms for manipulating compressed images" by Brian C. Smith, Lawrence A. Rowe, IEEE Transactions on Computer Graphics and Applications, September, 1993, and in US patent 5751865 (Felice A. Micco, Martin E. Banton).

[0008] However, the techniques of compressing a digital signal based on a discrete cosine transform are less efficient in terms of compression than the compression techniques based on a spectral breakdown of the digital signal into frequency sub-bands.

[0009] The present information aims to remedy the drawbacks of the prior art by proposing a method and device for transcoding a compressed data file by a compression technique using a spectral breakdown of the digital signal.

[0010] To this end, the invention concerns a method for the geometric transcoding of a compressed data file containing a digital signal of dimension N coded by a coding method including at least one step of spectral breakdown into frequency sub-bands of the digital signal.

[0011] According to the invention, this transcoding method is characterised in that it includes the following steps:

- extraction of symbols associated with the coefficients of the frequency sub-bands in each direction of the digital signal;
- application of a geometric transformation to said symbols;
- updating of N indicator or indicators representing a normal or reversed order of the symbols respectively in N direction or directions of the digital signal; and
- reconstitution of the coded digital signal by reversal of the extraction step

[0012] Correlatively, the invention proposes a device for the geometric transcoding of a compressed data file containing a digital signal of dimension N coded by a coding method including at least one step of spectral breakdown into frequency sub-bands of the digital signal, characterised in that it has:

- means of extracting symbols associated with the coefficients of the frequency sub-bands in each direction of the digital signal;
- means of applying a geometric transformation to said symbols;
- means of updating N indicator or indicators representing a normal or reversed order of the symbols respectively in N direction or directions of the digital signal; and
- means of reconstituting the coded digital signal by reversal of the extraction step.

[0013] The geomtric transcoding method and device make it possible to effect a geometric transformation of a coded digital signal on the basis of a spectral breakdown without completely decompressing the digital signal since the precaution is taken of modifying an indicator intended to give information about the reversal or not of the frequency sub-band symbols, in each direction of the digital signal. This indicator makes it possible to adapt the subsequent decoding method, notably the spectral reconstruction of the digital signal as a function of the value of this indicator.

[0014] In addition, this transcoding method avoids making modifications to the value of the symbols associated with the coefficients of the frequency sub-bands of the digital signal, without any increase in the calculation quantity necessary at the time of decoding.

[0015] According to one advantageous characteristic of the invention, the extraction step includes a reading of the coded digital signal, the signals being entropic codes associated respectively with the coefficients of the frequency sub-bands obtained by spectral breakdown.

[0016] This embodiment is particularly advantageous since it requires a minimal decompression of the compressed data file, the geometric transformation being applied directly to the entropic codes contained in the compressed data file, referred to as prefix codes. It is, however, well suited only when the coding of the digital signal uses an entropic coding, of the Huffman coding type, which associates an entropic coding with each coefficient of the signal of the frequency sub-bands.

[0017] According to a preferred version of the invention, which minimises the decompression of the file and is applied for a large number of conventional coding methods, using a scalar quantization of the coefficients of the signal of the sub-bands, the extraction step includes an entropic decoding of the coded digital signal, the symbols being quantization symbols associated respectively with the coefficients of the frequency sub-bands obtained by spectral breakdown.

[0018] Alternatively, according to another preferred version, which avoids the complete decompression of the file, and notably the spectral recomposition of the digital signal, the extraction step includes a dequantization of the coded digital signal, the symbols being the dequantized coefficients of the frequency sub-bands obtained by spectral breakdown.

[0019] This embodiment is particularly well suited when the coding method uses a vector quantization of the digital signal broken down into frequency sub-bands.

[0020] According to a preferred version of the invention, the indicator is a supplementary bit recorded in the compressed data file, having an initial value representing a normal order of the coefficients of the frequency sub-bands in a direction associated with said indicator.

[0021] Such an indicator is particularly advantageous in terms of memory space used in the compressed data file. It is also very easily updated from its initial value, fixed for example at the time of coding of the digital signal which has not undergone any geometric transformation, for example by alternation of the values 0 and 1 representing respectively a normal or reversed order of the symbols, for each direction of the digital signal.

[0022] In a preferred implementation of the invention, which applies in particular to geometric manipulations of the images, for example, for printing them, the digital signal then being of dimension 2, the geometric transformation applied is a transformation by axial or central symmetry of said symbols, a transformation by rotation through a multiple of 90° or a combination of said transformations.

[0023] In this same preferred embodiment of the invention, the transcoding method also comprises a step of transposition of a frequency sub-band having coefficients of low frequency in a first direction of the digital signal and high frequency in a second direction of the digital signal with a frequency sub-band with the same resolution level in the spectral breakdown, having coefficients of high frequency in said first direction and low frequency in said second direction, when the geometric transformation applied comprises a rotation through 90° or 270°.

[0024] This characteristic of the transcoding method makes it possible to take into account the changes in direction of the symbols during the rotations through 90° or 270° of an image for example, for a correct application of the method of decoding the digital signal.

[0025] According to another preferred version of the invention, the compressed data file containing several digital signals sequenced in a pre-determined order, the transcoding method also includes a step of sequencing the digital signals according to the geometric transformation applied.

[0026] The transcoding method thus applies particularly well to a digital signal divided into digital subsignals before being coded, the subsignals being stored in a pre-determined order in the compressed data file.

[0027] The invention also concerns a method of decoding a compressed data file containing a digital signal of dimension N coded by a coding method including at least one step of spectral breakdown into frequency sub-bands of the digital signal, characterised in that it comprises, in each direction of the digital signal, the following steps:

- reading an indicator representing a normal or reversed state of coefficients of the frequency sub-bands in said direction of the digital signal;
- calculating the parity of the digital signal in said direction;
- transforming original spectral recomposition filters in said direction as a function of the parity of the digital signal and the value of the indicator; and
- spectral recomposition of the digital signal by means of transformed recomposition filters.

[0028] Correlatively, the invention also concerns a device for decoding a compressed data file containing a digital signal of dimension N coded by a coding method including at least one step of spectral breakdown into frequency sub-bands of the digital signal, characterised in that it has:

- means of reading an indicator representing a normal or reversed state of coefficients of the frequency sub-bands

in a direction of the digital signal;

- means of calculating the parity of the digital signal in said direction;
- means of transforming original spectral recomposition filters in said direction as a function of the parity of the digital signal and the value of the indicator; and
- means of spectral recomposition of the digital signal by means of transformed recomposition filters.

[0029] This decoding method and device make it possible to decode a digital signal taking account of any reversal of the coefficients of the signal of the frequency sub-bands for the spectral reconstruction of the digital signal.

[0030] The transformation of the reconstruction filters can be effected in a relatively simple fashion, according to the parity of the digital signal and the reversal or not of the coefficient of the signal.

[0031] According to an advantageous version of the invention, which uses a relatively simple transformation of the filters, the time of the transformation of the original spectral recomposition filters, these filters are made symetrical and/or offset by an index.

[0032] According to a preferred version of the invention, particularly well adapted when the digital signal coding method comprises, in each direction of the signal, a processing of the start and a processing of the end of the digital signal, the decoding method according to the invention also includes the following steps:

- calculating the parity of the spectral recomposition filters;
- choosing the processing to be applied to said digital signal as a function of the parity of the spectral recomposition filters, the parity of the signal and the value of the indicator; and
- applying said chosen processing to the digital signal before the spectral recomposition step.

[0033] The decoding method thus makes it possible to take into account the edge problems which generally present themselves at the end of a finite monodirectional digital signal. It makes it possible to reverse the processing to be applied to the ends at the time of decoding if the coefficients of the sub-band signal have been reversed.

[0034] In an advantageous version of the invention, particularly simple to implement, the digital signal start and end processings are symetrical extensions of the digital signal.

[0035] According to a preferred version of the invention, the decoding method also includes a prior step of dividing the compressed data file into several digital signals sequenced in a pre-determined order, the size of said signals being determined according to the value of the indicators.

[0036] The decoding method thus takes account also of the change in the order of signals made at the time of transcoding of the signals, so as to redivide the transcoded digital signal into sub-signals corresponding in size to the sub-signals divided at the time of coding of the digital signal.

[0037] The invention also concerns a method of coding a digital signal of dimension N adapted to be transformed geometrically by a transcoding method according to the invention, characterised in that it comprises the following steps;

- spectral breakdown into frequency sub-bands of the digital signal; and
- entry into a compressed file comprising the coded digital signal of N indicator or indicators associated respectively with N direction or directions of the digital signal, in the form of a supplementary bit having an initial value representing a normal order of the coefficients of the frequency sub-bands in a direction associated with said indicator.

[0038] Correlatively, the invention concerns a device for coding a digital signal of dimension N adapted to be transformed geometrically by a transcoding method according to the invention, characterised in that it has:

- means of spectral breakdown into frequency sub-bands of the digital signal; and
- means of entry into a compressed file containing the coded digital signal of N indicator or indicators associated respectively with N direction or directions of the digital signal, in the form of a supplementary bit having an initial value representing a normal order of the coefficients of the frequency sub-bands in a direction associated with said indicator.

[0039] This coding method and device has the advantage of associating with the digital signal, as from coding, an indicator representating a normal, non-transformed state of the digital signal.

[0040] By associating an indicator with each direction of the digital signal, it is then possible, at the time of transcoding, to modify these indicators in order to take account of the transformations of the digital signal, in a compressed form, in all the directions of the space containing the digital signal.

[0041] The invention also relates to a digital signal processing apparatus having means adapted to implement the transcoding method, the decoding method or the coding method, or again having a transcoding device, a decoding device or a coding device as disclosed above.

[0042] The transcoding, decoding and coding methods according to the invention are particularly well adapted to be used in a digital photographic apparatus, a computer or a photocopier.

[0043] Correlatively, the invention also concerns a computer, a photocopier or a digital photographic apparatus having a transcoding device, a decoding device or a coding device according to the invention.

[0044] The transcoding and decoding methods are also particularly well adapted to be used in a digital printer.

[0045] Correlatively, the invention also concerns a digital printer comprising a transcoding device or a decoding device according to the invention.

[0046] The advantages of the digital signal processing apparatus, the digital photographic apparatus, the computer, the photocopier and the digital printer are similar to those of the methods which they implement or the devices which they include.

[0047] An information storage means, which can be read by a computer or a microprocessor, integrated or not into it, possibly removable, stores a program implementing the coding and/or transcoding and/or decoding method.

[0048] In this way, the present invention concerns a computer program product loadable into a computer or stored on a computer usable medium comprising software code portions for performing the steps of the coding and/or transcoding and/or decoding method when it runs on a computer.

[0049] Other particularities and advantages of the invention will also emerge from the following description of a preferred embodiment of the invention.

[0050] In the accompanying drawings, given by way of non-limitative example:

- Figure 1 is a block diagram of a coding device according to one embodiment of the invention;
- Figure 2 is a block diagram of a transcoding device according to one embodiment of the invention;
- Figure 3 is a block diagram of a decoding device according to an embodiment of the invention;
- Figure 4 is a circuit for breaking down into frequency sub-bands and for recomposition, included in the coding and decoding devices illustrated respectively in figures 1 and 3;
- Figures 5 and 6 illustrate a spectral breakdown of an image into frequency sub-bands;
- Figure 7 depicts a device for processing a digital signal adapted to implement the coding, transcoding and decoding methods of the invention;
- Figure 8 is a coding signal coding algorithm according to one embodiment of the invention;
- Figure 9 is a transcoding algorithm according to one embodiment of the invention;
- Figure 10 is a decoding algorithm according to one embodiment of the invention;
- Figure 11 illustrates schematically the transcoding method applied to several sequenced digital signals; and
- Figure 12 is an algorithm for transcoding several sequenced digital signals.

[0051] A description will first of all be given, with reference to Figure 1, of a coding device according to one embodiment of the invention intended to code a digital signal S for the purpose of compressing it.

[0052] In this particular embodiment, the digital signal S is of dimension 2, formed by a series of digital samples representing an image. The digital samples are for example bytes, each byte value representing a pixel of an image, here with 256 grey levels, or a black and white image.

[0053] The signal S is supplied to the coding device in a convention fashion by a signal source which either contains the digital signal and is for example a memory, a hard disk or a CD-ROM, or converts an analogue signal into a digital signal, and is for example an analogue photographic apparatus associated with an analogue to digital convertor.

[0054] The coding device has means 2 for the spectral breakdown into frequency sub-bands of the digital signal.

[0055] These spectral breakdown means consist of a conventional set of filters, respectively associated with decimators by two, which filter the image in two directions.

[0056] As the filtered signal has a support wider than the support of the original signal, the coding device conventionally has processing means 1, upstream of the breakdown means 2, adapted to process the ends of the digital signal in order to avoid the problems related to edge effects.

[0057] In this example embodiment, the processing means 1 for the start and end of a digital signal are adapted to produce symetrical extensions of the digital signal. These processing means 1 can be adapted to implement any other equivalent processing, for example, aliasing of the filters.

[0058] In the usual fashion, a digital signal of length K, X(0) X(1) X(2) X(3).........X(K-4) X(K-3) X(K-2) X(K-1) can be extended at its ends by:

an extension whose centre of symmetry is the last sample of the signal (extension P):
X(2) X(1)/ X(0) X(1) X(2) X(3)......X(K-4) X(K-3) X(K-2) X(K-1)/ X(K-2) X(K-3)

or

- an extension whose centre of symmetry is the half-sample external to the signal (extension 1/2P):
X(1) X(0)/ X(0) X(1) X(2) X(3).....X(K-4) X(K-3) X(K-2) X(K-1)/ X(K-1) X(K-2)

or

- an extension whose centre of symmetry is the last sample of the signal and where the samples external to the signal are multiplied by -1 (extension -P):
  -X(2) -X(1)/ X(0) X(1) X(2) X(3)...X(K-4) X(K-3) X(K-2) X(K-1)/ -X(K-2) -X(K-3)
  or
- an extension whose centre of symmetry is the half-sample external to the signal and where the samples external to the signal are multiplied by -1 (extension -1/2P):
  -X(1) -X(0)/ X(0) X(1) X(2) X(3)...X(K-4) X(K-3) X(K-2) X(K-1)/ -X(K-1) -X(K-2)
  or
- an extension whose centre of symmetry is a zero at the ends of the signal and where the samples external to the signal are multiplied by -1 (extension -1/2P0):
  X(1) -X(0) 0X(0) X(1) X(2) X(3)...X(K-4) X(K-3) X(K-2) X(K-1) 0 -X(K-1) - X(K-2)

[0059] The type of extension used is identical at the two ends of the digital signal at the time of coding and spectral breakdown thereof. On the other hand, the extension can be different at the time of decoding and spectral recomposition of the signal, according to the parity of the signal and the filter used. This modification of the extension type will be explained in more detail below, in relation to the decoding method according to the invention.

[0060] In this example embodiment, the means 1 of processing the ends of the signal are adapted to use an extension P if the length parity of the breakdown filters used is odd and an extension 1/2P if the length parity of the filters is even.

[0061] Preferably, the spectral breakdown means 2 are adapted to effect a breakdown into discrete wavelets and consist of a circuit for decomposing into sub-bands, or analysis circuit, formed by a set of analysis filters 21, 22, respectively associated with decimators by two 210, 220, (see Figure 4). This breakdown circuit filters the image signal S in two directions, into sub-bands of spatial high frequencies and low frequencies.

[0062] These low-pass h1(k) and high-pass g1(k) breakdown filters have the same length parity, herein referred to as the parity of the filters.

[0063] In this example, only one analysis unit has been depicted. The breakdown circuit does however preferably have several successive analysis units for breaking the signal S down into sub-bands according to several resolution levels.

[0064] This breakdown into sub-bands is well known and the different analysis steps used will be briefly stated below, with reference to Figures 5 and 6, in the case of an image I broken down into sub-bands at a breakdown level equal to 3.

[0065] A first analysis unit receives the image signal I and filters it through two digital filters, respectively low-pass and high-pass, in a first direction, for example horizontal. After passing through decimators by two, the resulting filtered signals are in turn filtered by two filters respectively low-pass and high-pass, in a second direction, for example vertical. Each signal is once again passed through a decimator by two. This then gives, at the output of this first analysis unit, four sub-bands LL_1, LH_1, HL_1 and HH_1 with the highest resolution in the breakdown.

[0066] The sub-band LL_1 includes the components of low frequency in the two directions of the image signal I. The sub-band LH_1 includes the components of low frequency in a first direction and of high frequency in a second direction of the image signal I. The sub-band HL_1 includes the components of high frequency in the first direction and the components of low frequency in the second direction. Finally, the sub-band HH_1 includes the components of high frequency in both directions.

[0067] A second analysis unit in its turn filters the sub-band LL_1 in order to supply, in the same way, four sub-bands LL_2, LH_2, HL_2 and HH_2 with an intermediate resolution level in the breakdown. Finally, in this example, the sub-band LL_2 is in its turn analysed by a third analysis unit in order to provide four sub-bands LL_3, LH_3, HL_3 and HH_3 with the lowest resolution in this breakdown.

[0068] Thus ten sub-bands and three resolution levels are obtained. Naturally, the number of resolution levels, and consequently of sub-bands, can be chosen differently, and can for example be equal to four resolution levels with thirteen sub-bands.

[0069] In another embodiment, the sub-bands HL, LH and HH can also be broken down into sub-bands of lower resolution.

[0070] As illustrated in Figure 1, the coding device also has quantization means 3 for quantizing, in a conventional fashion, by means of quantization symbols, the coefficients of the signal broken down into frequency sub-bands, and entropic coding means 4 for coding the quantization symbols.

[0071] These different means of processing 1, spectral breakdown 2, quantization 3 and entropic coding 4 are in normal use in the field of image comparison and will not be described in any more detail here.

[0072] In this preferred embodiment, the quantization means 3 are adapted to perform a scalar quantization of the coefficients of the signal of the sub-bands LL_3, LH_3, HL_3, HH_3, LH_2, HL_2, HH_2, LH_1, HL_1 and HH_1 Thus each coefficient c_i of the frequency sub-bands is depicted in a unique fashion by a quantization symbol q_i. By way of example, the entropic coding can be an arithmetic coding.

[0073] According to the invention, the coding device has means 5 of entering in a file compressed data F containing the coded digital signal of N indicator or indicators associated with each direction of the digital signal, N here being equal to 2.

[0074] This indicator is in the form of a supplementary bit having an inital value representing a normal order of the coefficients of the signal of the frequency sub-bands in a direction associated with the indicator.

[0075] For an image of dimension 2, two indicators are entered in the compressed file F, the first indicator Ch associated with the horizontal direction and a second indicator Iv associated with the vertical direction of the digital signal S.

[0076] The initial value representing the normal order of the sub-band coefficients, that is to say of an image which has not undergone any geometric transformation, can be equal to 0 by way of example.

[0077] In this way, there is obtained at the output of the coding device a compressed data file F comprising the digital signal S coded and not transformed geometrically, and the indicators Ih and Iv entered with their initial value 0.

[0078] A description will now be given, with reference to Figure 2, of the transcoding device according to one embodiment of the invention, adapted to transcode a compressed file F by means of the coding device described above.

[0079] This geometric transcoding device has:

- means 6 of extracting symbols associated with the coefficients of the frequency sub-bands in each direction of the digital signal;
- means 7 of applying a geometric transformation to the symbols;
- means 8 of updating the indicators Ih and Iv representing a normal or reversed order of the symbols respectively in the two directions of the digital signal; and
- means 10 of reconstituting the coded digital signal transformed geometrically in the compressed file F*.

[0080] In this example, and in relation to the embodiment described above of the coding device, the extraction means 6 are adapted to perform an entropic decoding of the coded digital signal, the symbols being quantization symbols q_i associated respectively with the coefficients c_i of the signal of the frequency sub-bands obtained by spectral breakdown.

[0081] The means 7 of applying a geometric transformation are for example adapted to perform a transformation of the symbols q_i by axial or central symmetry, a transformation by rotation by a multiple of 90° or a combination of these transformations. They are adapted to update the width and height of the image in the compressed file, the height and width being reversed when the image is pivoted through 90° or 270°.

[0082] The decoded image can thus, by way of example, be transformed according to an axial symmetry (vertical or horizontal), be turned over (rotation through 180°) or oriented differently (rotation through 90° or 270°).

[0083] The means 8 of updating the indicators Ih and Iv modify the value of this indicator when the transformation applied modifies the order of the symbols in the direction associated with the indicator. The value of the indicator is then equal to 1 when the order of the symbols is reversed with respect to the decomposition.

[0084] Table 1 below illustrates, according to the different types of geometric transformation applied, the new values, denoted I'h and I'v, of the indicators Ih and Iv, which replace the old values in the transcoded compressed file.

[0085] These new values I'h and I'v are a function of the geometric transformation applied and the old values of the indicators Ih and Iv.

[0086] The bit which is the inverse of the bit I is hereinafter denoted inv(I), that is to say inv(I) = 1 when I = 0 and inv (I) = 0 when I = 1.

[0087] The rotations are indicated in the clockwise direction.

Table 1

Transformation	I'h	I'v
Identity	Ih	Iv
rotation through 90 degrees	inv(Iv)	Ih
rotation through 180 degrees	inv(Ih)	inv(Iv)
rotation through 270 degrees	Iv	inv(Ih)
vertical symmetry axis	inv(Ih)	Iv
vertical symmetry axis then rotation through 90 degrees	inv(Iv)	inv(Ih)
vertical symmetry axis then rotation through 180 degrees	Ih	inv(Iv)
vertical symmetry axis then rotation through 270 degrees	Iv	Ih

[0088] The reconstitution means 10 are in this example entropic coding means adapted to recompress the data file

containing the digital signal which was transformed geometrically.

[0089] In this example embodiment in dimension 2, the transcoding device also comprises means 9 of transposing the frequency sub-bands HL_3 HL_2 and HL_1 having coefficients c_i of low frequency in the first direction of the digital signal and of high frequency in a second direction of the digital signal with respectively of the frequency sub-bands LH_3, LH_2 and LH_1 with the same resolution level in the spectral breakdown, having coefficients c_i of high frequency in the first direction and low frequency in the second direction. This transposing of the sub-bands is effected when the geometric transformation applied comprises a rotation through 90° or 270° which modifies the direction of the coefficients c_i.

[0090] In this way there is obtained at the output of the transcoding device a compressed data file F* which is transcoded with respect to the initial compressed data file F so that, on decoding, the decoded digital signal S* will have been transformed geometrically with respect to the initial digital signal S.

[0091] A description will now be given of the decoding device according to one embodiment of the invention, with reference to Figure 3, which makes it possible to decode a compressed data file transcoded by means of the transcoding device described previously.

[0092] This decoding device has means 11 of reading an indicator Iv or Ih representing a normal or reversed state of coefficients c_i of the signal of the frequency sub-bands in one direction of the digital signal.

[0093] It also has entropic decoding means 12 and dequanization means 13 for decoding and dequantizing the compressed digital signal in a conventional fashion.

[0094] According to the invention, it also comprises means 14 of calculating the parity of the digital signal and means 18 of transforming the original spectral recomposition filters as a function of the parity of the digital signal and the value of the indicator.

[0095] Means 19 for the spectral recomposition of the digital signal are adapted to reconstruct the signal by means of transformed recomposition filters.

[0096] The spectral recomposition means 19 comprise a synthesis circuit which corresponds to the analysis circuit described in the coding device. In a known fashion, and as illustrated in Figure 4, synthesis filters 24 and 25 are associated respectively with multipliers by two 240 and 250. The synthesis circuit of course has as many synthesis units as the analysis circuit has analysis units. In this example embodiment, the synthesis unit will have three successive synthesis units.

[0097] It is known that the conditions for perfect reconstruction of the digital signal entail the parity of the low-pass h2(k) and high-pass g2(k) recomposition filters being identical to that of the breakdown filters h1(k) and g1(k).

[0098] In addition the low-pass breakdown hk(1) and low-pass recomposition h2(k) recomposition filters are symmetrical.

[0099] In general terms, the breakdown and recomposition filters verify the following equations, which afford perfect reconstruction of the digital signal S:

$$g1(k) = (-1)^k h2(-k+1)$$

and

$$g2(k) = (-1)^k h1(-k+1)$$

[0100] The transformation means 18 of the recomposition filters h2 and g2 are adapted to modify these filters in order to take account of the reversals in the order of the coefficients c_i in the sub-bands.

[0101] Table 2 below illustrates the transformation of the inital synthesis filters, which can be interpreted as the construction of new filters h'2 and g'2:

Table 2

Parity of signal to be reconstructed	Ih (ou Iv)	h'2(k)	g'2(k)
Odd	0	h2(k)	g2(k)
Odd	1	h2(-k)	g2(-k)
Even	0	h2(k)	g2(k)
Even	1	h2(-k+1)	g2(-k+1)

[0102] The transformation means 18 of the original spectral recomposition filters are thus adapted to make the spectral recomposition filters h2(k) and g2(k) symmetrical or to shift them by an index.
[0103] The transformation of the filters can, for a simplification of the implementation of the decoding device, also be interpreted as a conservation of the original recomposition filters and a modification of their application. In this interpretation, the synthesis filters h2 and g2 are used systematically. If the filters are of even length, the opposite of the high-pass signal is taken to effect the spectral recomposition. In addition, if the digital signal is of even length and the indicator Ih (or Iv) is equal to 1, the over-sampling performed at the time of the spectral recomposition is adapted to insert the first zero before the first coefficient of the signal (whereas in a conventional spectral recomposition the first zero is inserted after the first coefficient).
[0104] According to the invention the decoding device also has :

- means 15 of calculating the parity of the spectral recomposition filters h2(k) and g2(k);
- means (16) of choosing the processing to be applied to the digital signal according to the parity of the spectral recomposition filters h2(k) and g2(k), the parity of the signal S and the value of the indicator Iv or Ih; and
- means 17 of applying the chosen processing to the digital signal.

[0105] This is because, as known in a spectral breakdown, when a processing A and a processing B are respectively applied to the ends of a signal before its spectral breakdown, it is necessary to apply, before its reconstruction, a processing A' and a processing B', the choice of these processings depending on the parity of the filters and the parity of the signal to be reconstructed.
[0106] The means of choosing 16 and applying 17 the processing according to the invention make it possible to reverse the end processings when the coefficients of the signal of the sub-bands have been reversed.
[0107] Two tables are given below, by way of example, for choosing the processing to be applied to the ends of the signal when, on coding, a processing of type P is chosen when the parity of the filters is odd and type 1/2P when the parity of the filters is even.
[0108] Table 3 gives the extensions of the signal for its reconstruction by the low-pass recomposition filter.
[0109] Table 4 given the extensions of the signal for its reconstruction by the high-pass recomposition filter.

Table 3

Parity of filters	Parity of signal	Ih (or Iv)	extension of start of signal	extension of end of signal
Odd	Odd	0	P	P
Odd	Odd	1	P	P
Odd	Even	0	P	1/2P
Odd	Even	1	1/2P	P
Even	Odd	0	1/2P	P
Even	Odd	1	P	1/2P P
Even	Even	0	1/2P	1/2P
Even	Even	1	1/2P	1/2P

Table 4

Parity of filters	Parity of signal	Ih (or Iv)	extension of start of signal	extension of end of signal
Odd	Odd	0	1/2P	1/2P
Odd	Odd	1	1/2P	1/2P
Odd	Even	0	1/2P	P
Odd	Even	1	P	1/2P P
Even	Odd	0	-1/2P	-1/2P0
Even	Odd	1	-1/2P0	-1/2P
Even	Even	0	-1/2P	-1/2P
Even	Even	1	-1/2P	-1/2P

[0110] In a preferred embodiment of the invention, the means of processing 1, breakdown 2, quantization 3 and entropic coding 4 and the recording means 5 for the coding device are incorporated in a microprocessor 100, a read-only memory 102 containing a program for coding the digital signal, and a random access memory 103 containing registers adapted to record variables modified during the running of said program.

[0111] In a similar fashion, the means of extracting 6, applying 7 a geometric transformation, updating 8, transposing 9 and of reconstituting 10 of the transcoding device are incorporated in a microprocessor 100, a read-only memory 102 containing a program for geometrically transcoding a coded digital signal, and a random access memory 103 containing registers adapted to record variables modified during the running of the program.

[0112] Likewise, the means of reading 11, entropic decoding 12, dequantization 13, calculation 14 of the parity of the digital signal, calculation 15 of the parity of the filters, choosing 16, applying 17, transforming 18 and spectral recomposition 19 of the decoding device are incorporated in the microprocessor 100, a read-only memory 102 containing a program for decoding the coded digital signal, and a random access memory 103 containing registers adapted to record variables modified during the running of the program.

[0113] The microprocessor 100, the read-only memory 102 and the random access memory 103 can be integrated into a computer 20 as illustrated in Figure 7, connected to different peripherals, for example a digital camera 107 (or a scanner, or any other image acquisition or storage means), connected to a graphics card and supplying data to be coded, transcoded or decoded according to the present invention.

[0114] The computer 20 has a communication interface 112 connected to a network 113 able to transmit digital information to be coded by the computer or to transmit compressed digital information to be transcoded or decoded by the computer. Conversely, the network 113 can transmit digital information from the computer 20.

[0115] The computer 20 also has a storage means 108 such as for example a hard disk. It also has a disk drive 109 adapted to read a diskette 110. The diskette 110 and the hard disk 108 can contain digital data to be processed according to one of the methods according to the invention, and the code of the invention which, once read by the computer 20, will be stored on the hard disk 108.

[0116] According to a variant, the program for implementing the coding, transcoding and decoding methods of the invention can be stored in a read-only memory 102. According to a second variant, this program can be received in order to be stored in a read-only memory 102 by means of the communication network 113.

[0117] The computer 20 is connected to a microphone 111 by means of an I/O card 103. The processed data will in this case by an audio signal of dimension 1.

[0118] The computer 20 has a screen 104 for displaying the information to be processed or to serve as an interface with the user, who will be able to parametrize certain coding, transcoding (applied geometric transformation type for example) or decoding modes, by means of the keyboard 114 or any other means (a mouse for example).

[0119] The microprocessor 100 will execute the instructions relating to the implementation of the invention, instructions stored in the read-only memory 102 or in other storage elements. On powering up, the programs and methods stored in one of the non-volatile memories are transferred into the random access memory 103, which will then contain the executable code of the invention. In a variant, the coding, transcoding and decoding methods can be stored in different places. This is because it is possible to improve the methods according to the invention by adding new methods transmitted either by the communication network 113 or by a diskette 110.

[0120] Naturally, the diskettes can be replaced by any data medium such as a CD-ROM or a memory card.

[0121] The communication bus 101 affords communication between the different elements of the computer 20 or those connected to it. The representation of the bus 101 is not limitative and notably the central unit 100 is able to communicate instructions to any sub-element of the computer 20 directly or by means of another sub-element of the computer 20.

[0122] The computer described above can contain all or part of the coding device according to the invention. It can also contain all or part of the transcoding device according to the invention. Finally, it can contain all or part of the decoding device according to the invention.

[0123] A description will now be given successively of the methods of coding, transcoding and decoding a digital signal S with reference to Figures 8, 9 and 10.

[0124] The coding method comprises the following steps E1 to E8:

[0125] First of all, in a step E1, the digital signal S is considered in a first direction, for example horizontal for a signal of dimension 2 representing a digital image.

[0126] According to the invention, a recording step E2 makes it possible to write, in the compressed file F comprising the coded digital signal, an indicator associated with the first direction of the digital signal. This indicator, here Ih, can be in the form of a supplementary bit having an initial value, for example 0, representing a normal order of the coefficients of the sub-band signal in this horizontal direction.

[0127] A test is next carried out, at a step E13, to determine whether all the directions of the digital signal have been envisaged.

[0128] In the negative, in a step E4, another direction of the digital signal is considered, here the vertical direction,

and the step E2 of recording an indicator, here Iv, in the compressed file F, is reiterated.

[0129] The ends of the digital signal are next processed in a processing step E5, for example by means of a symmetrical extension of type P.

[0130] A step E6 of spectral breakdown into frequency sub-bands of the digital signal is next implemented in a conventional fashion and as described above, for example, at a breakdown level equal to 3.

[0131] Each sub-band LL_3, LH_3, HL_3, HH_3, LH_2, HL_2, HH_2, LH_1, HL_1 and HH_1 is then quantized in a quantization step E7 and then coded entropically in an entropic coding step E8.

[0132] The digital signal thus coded is stored in a compressed data file F with a view to its storage or transmission.

[0133] When a geometric transformation of the digital signal is to be applied, the compressed file F is transcoded according to the transcoding method as illustrated in Figure 9 and which comprises the following steps:

[0134] In an extraction step E9, symbols associated with the coefficients c_i of the frequency sub-bands of the digital signal are extracted.

[0135] In this example of a preferred embodiment, the extraction step E9 corresponds to an entropic decoding of the coded digital signal in order to extract the quantization symbols q_i associated with each coefficient c_i.

[0136] Next a geometric transformation step E10 is applied to the symbols q_i, if applicable modifying their order in each direction of the signal. The geometric transformation applied is a transformation by axial or central symmetry of the symbols, a transformation by a rotation by a multiple of 90° or a combination of these transformations.

[0137] In addition, where the quantization step E7 includes a quantizer which varies according to a direction of the digital signal, in at least one sub-band of the digital signal, a geometric transformation identical to this quantizer is applied. For example, if the quantizer is associated with a quantization step table, the geometric transformation is applied to this quantization step table.

[0138] Next, in a step E11, a direction of the digital signal, for example horizontal, is considered.

[0139] A step E12, E13 of updating the indicator, here Ih, representing a normal or reversed order of the symbols in the horizontal direction, is next implemented.

[0140] For example, it is checked, in a test step E12, whether the direction of the symbols q_i is reversed. This check can be carried out from a table such as Table 1 associating with each geometric transformation, chosen for example by a user, a value 0 or 1 representing the normal or reversed order of the symbols q_i in the horizontal direction.

[0141] If the order is reversed, in a modification step E13, the value of the indicator Ih is modified. Otherwise, the latter remains unchanged.

[0142] A test step E14 checks whether all the directions have been envisaged.

[0143] In the negative, another direction of the signal is envisaged in a step E15, here the vertical direction, and the set of steps E12 and E13 is reiterated.

[0144] It is next checked, in a test step E16, whether the geometric transformation applied comprises a rotation through 90° or 270°.

[0145] In the affirmative, the transcoding method comprises a step E17 of transposing of the frequency sub-bands HL_3, HL_2, HL_1 having coefficients of low frequency in a first direction of the digital signal and of high frequency in a second direction of the digital signal with respectively the frequency sub-band LH_3, LH_2, LH_1 of the same resolution level in the spectral breakdown, having coefficients of high frequency in this first direction and of low frequency in this second direction.

[0146] Next, in a reconstitution step E18, the complete recompression of the coded digital signal is carried out, by reversing the extraction step E9. In this embodiment, the quantization symbols q_i transformed geomtrically by an entropic coding method similar to that used in the entropic coding step E8 of the coding method are therefore coded.

[0147] The transcoding method is then terminated and the transformed compressed file F* is stored or transmitted for subsequent processing.

[0148] When it is wished to extract the image from the compressed file F*, this is decoded by the decoding method according to the invention illustrated in Figure 10.

[0149] This decoding method includes first of all, in a conventional fashion, an entropic decoding step E19, the reverse of the entropic coding step E8 of the coding method, and a dequantization step E20, the reverse of the quantization step E7 of the coding method.

[0150] A first direction of the signal is next envisaged in a step E21, for example the horizontal direction.

[0151] A reading step E22 reads the indicator Ih representing a normal or reversed state of the coefficients of the frequency sub-bands in the horizontal direction of the digital signal.

[0152] In a step E23 it is tested whether the order of the coefficients is reversed, for example by checking whether the value of Ih is equal to 1.

[0153] In the affirmative, a step E24 of making filters symmetrical, as illustrated in Table 2, is performed.

[0154] In a calculation step E25, the parity of the digital signal in the horizontal direction is calculated.

[0155] In a step E26, it is checked whether the digital signal to be reconstructed is even and, in the affirmative, still when the order of the coefficients is reversed, a shifting by an index of the spectral recomposition filters is also carried

out as illustrated on the last line of Table 2.

[0156] The steps of making symmetrical E24 and shifting E27 thus correspond to a step of transforming the original spectral recomposition filters in the horizontal direction according to the parity of the digital signal or the value of the indicator Ih.

[0157] In a step E28, a first recomposition filter is next considered, for example the low-pass recomposition filter h'2 transformed as described above using the initial recomposition h'2.

[0158] Whether or not the order of the coefficients is reversed, the parity of the spectral recomposition filter h'2 is next calculated is a calculation step E29 and, in a choosing step E30, the processing to be applied to the digital signal as a function of the parity of the spectral recomposition filter, the parity of the signal and the value of the indicator Ih is chosen. This choice can be made for example using Table 3.

[0159] A step E31 of applying the chosen processing is next implemented on the digital signal to be reconstructed.

[0160] In step E32, it is tested whether all the recomposition filters have been envisaged and, in the negative, in a step E33, the following filter is considered, here the high-pass recomposition filter g'2. Steps E29 to E32 are reiterated for the processing of the ends of the digital signal, using in this case, for the choice of the symmetrical extension processing to be applied, Table 4.

[0161] Next, in a test step E34, it is checked whether all the directions of the digital signal have been envisaged and, in the negative, the following direction, here the vertical direction, is considered at step E35, so that steps E22 to E34 of the decoding method are reiterated.

[0162] The decoding method next includes a step E36 of the spectral recomposition of the digital signal by means of the transformed recomposition filters h'2 and g'2.

[0163] A digital signal S* which has undergone a geometric transformation with respect to the initial digital signal S is obtained at the output.

[0164] Figure 11 illustrates an application of the transcoding method according to the invention to a compressed data file F containing several digital signals A, B, C, D sequenced in a predetermined order.

[0165] Such is the case notably when an initial digital signal, for example an image, is divided into signal sub-elements, each signal sub-element then being coded independently, and then stored in a predetermined order in the compressed data file to allow subsequent reconstruction of the initial digital signal.

[0166] Each subsignal A, B, C and D is transcoded separately into subsignals A', B', C' and D', these transcoded subsignals next being resequenced according to the geometric transformation applied. Thus the compressed data file F, containing a successive sequence of signals A, B, C and D becomes, at the end of the transcoding, a compressed data file F*, containing the successive sequence of signals B', A', D' and C'.

[0167] As illustrated in Figure 12, in this particular embodiment, the transcoding method also includes a step E40 of sequencing the digital signals A, B, C, D according to the geometrical transformation applied.

[0168] In an extraction step E37, a first subsignal is extracted from the file F, to which there are applied all the steps E9 to E17 of the transcoding method as described previously with reference to Figure 9.

[0169] In a test step E38, it is checked whether all the subsignals have been envisaged and, in the negative, at step E39, the following subsignal is considered and the transcoding steps E9 to E17 are reiterated.

[0170] When all the subsignals have been transcoded, these subsignals are resequenced in the sequencing step E40.

[0171] The reconstitution step E18 is next implemented in order to reconstitute the compressed data file F*.

[0172] For implementing this transcoding method in this embodiment, and as illustrated in Figure 2, the transcoding device also has sequencing means 30. These sequencing means 30 can be incorporated in a microprocessor 100 as illustrated in Figure 7.

[0173] On decoding, and as illustrated in Figure 10, the decoding method also includes a prior step E41 of dividing the compressed data file F* into several digital signals sequenced in a predetermined order, the size of the signal being determined according to the value of the indicators Ih, Iv.

[0174] This is because, when on decoding the initial digital signal has been divided into subsignals and this division is different at the ends of the signal, the choice of the division to be carried out in order to redivide the transcoded signal into subsignals of identical size to that of the subsignals divided on decoding is made according to the value of the indicators Ic, Iv in each direction of the signal.

[0175] In order to implement this decoding method, the decoding device illustrated in Figure 3 also has dividing means 31. These dividing means 31 can be incorporated into a microprocessor 100 as illustrated in Figure 7.

[0176] The present invention thus makes it possible to transform a digital signal geometrically, avoiding as far as possible the phases of decompression and recompression of the digital signal when the latter is coded by a method using a multi-resolution spectral breakdown, of the wavelet type. In this way the number of calculations necessary and the memory space necessary for decompression of the signal are substantially reduced.

[0177] Thus, for example, it may be convenient, when it is wished to print a digital image, to send the latter to the printer in compressed form. The user can generally choose between "portrait" or "landscape" print modes. It suffices,

by virtue of the invention, according to the mode chosen, to effect a rotation of the compressed file through 90°, and then to print.

[0178] Likewise, digital photographic appliances generally have a "diaporama" mode in which all the images are displayed one after the other on a screen, all in the same direction. However, the user takes at least two types of photograph, some with the apparatus horizontal, the others with the apparatus vertical. In effecting the diaporama, certain images will be displayed askew. The transcoding method according to the invention, associated with the coding and decoding methods, makes it possible to effect a rotation on the coded images before they are displayed.

[0179] Naturally, numerous modifications can be made to the example embodiment described above without departing from the scope of the invention.

[0180] Thus the extraction means 5 of the transcoding device could be adapted to directly read the coded digital signal, the symbols transformed by the series being entropic codes associated respectively with the coefficients C_i of the signal sub-band which are obtained by spectral breakdown. These entropic codes can be used when the entropic coding used at the time of coding of the digital signal is for example a Huffman coding.

[0181] Conversely, if the quantization signals used on coding the signal do not associate a quantization symbol q_i with each coefficient c_i, the extraction means 5 are adapted to effect a dequantization of the coded digital signal, the symbols to which the geometric transformation is applied then being the dequantized coefficients c_i of the sub-band signal obtained by spectral breakdown. Such extraction means are in particular used when the coding device uses a technique of vector quantization of the digital signal broken down into frequency sub-bands.

[0182] Although the example described concerns an image of dimension 2, the invention also applies to digital signals of dimension 1, 3 or more.

Claims

1. Method for the geometric transcoding of a compressed data file (F) containing a digital signal (S) of dimension N coded by means of a coding method including at least one step (E6) of spectral breakdown into frequency sub-bands of the digital signal (S), characterised in that it includes the following steps:

   - extraction (E9) of symbols associated with the coefficients (c_i) of the frequency sub-bands in each direction of the digital signal (S);
   - application (E10) of a geometric transformation to said symbols;
   - updating (E12, E13) of N indicator or indicators ((Ih, Iv) representing a normal or reversed order of the symbols respectively in N direction or directions of the digital signal (S); and
   - reconstitution (E18) of the coded digital signal by reversal of the extraction step (E9).

2. Transcoding method according to Claim 1, characterised in that the extraction step (E9) includes a reading of the coded digital signal, the symbols being entropic codes associated respectively with the coefficients (c_i) of the frequency sub-bands obtained by spectral breakdown.

3. Transcoding method according to Claim 1, characterised in that the extraction step (E9) includes an entropic decoding of the coded digital signal, the symbols being symbols of quantizations (q_i) associated respectively with the coefficients (c_i) of the frequency sub-bands obtained by spectral breakdown.

4. Transcoding method according to Claim 1, characterised in that the extraction step (E9) includes a dequantization of the coded digital signal, the symbols being the dequantized coefficients (c_i) of the frequency sub-bands obtained by spectral breakdown.

5. Transcoding method according to one of Claims 1 to 4, characterised in that the indicator (Ih, Iv) is a supplementary bit recorded in the compressed data file (F), having an initial value (0) representing a normal order of the coefficients (c_i) of the frequency sub-bands in a direction associated with said indicator (Ic, Iv).

6. Transcoding method according to one of Claims 1 to 5, the digital signal (S) being of dimension 2, characterised in that the geometric transformation applied is a transformation by axial or central symmetry of said symbols, a transformation by rotation by a multiple of 90° or a combination of said transformations.

7. Transcoding method according to Claim 6, characterised in that it also comprises a step (E17) of transposing a frequency sub-band (HL_3 HL_2 HL_1) having coefficients (c_i) of low frequency in a first direction of the digital signal and of high frequency in a second direction of the digital signal with a frequency sub-band (LH_3, LH_2, LH_1) with

the same resolution level in the spectral breakdown, having coefficients (c_i) of high frequency in said first direction and of low frequency in said second direction when the geometric transformation applied comprises a rotation through 90° or 270°.

8. Transcoding method according to one of Claims 1 to 7, the compressed data file (F) containing several digital signals (A, B, C, D) sequenced in a predetermined order, characterised in that it also includes a step (E40) of sequencing said digital signals (A, B, C, D) according to the geometric transformation applied.

9. Method of decoding a compressed data file (F*) containing a digital signal (S) of dimension N coded by means of a coding method including at least one step (E6) of spectral breakdown into frequency sub-bands of the digital signal (S), characterised in that it comprises, in each direction of the digital signal, the following steps:

- reading (E22) an indicator (Ih, Iv) representing a normal or reversed state of the coefficients (c_i) of the frequency sub-bands in said direction of the digital signal;
- calculating (E25) the parity of the digital signal (S) in said direction;
- transforming (E24, E27) original spectral recomposition filters (h2, g2) in said direction according to the parity of the digital signal (S) and the value of the indicator (Ih, Iv); and
- spectral recomposition (E36) of the digital signal (S*) by means of transformed recomposition filters (h'2, g'2).

10. Decoding method according to Claim 9, characterised in that, on transformation (E24, E27) of the original spectral recomposition filters (h2, g2), said filters are made symmetrical and/or shifted by an index.

11. Decoding method according to one of Claims 9 or 10, the method of coding the digital signal comprising, in each direction of said signal, a processing of the start and a processing of the end (E5) of the digital signal (S), characterised in that it also includes the following steps:

- calculating (E29) the parity of the spectral recomposition filters (h2, g2);
- choosing (E30) the processing of ends to be applied to said digital signal according to the parity of the spectral recomposition filters (h2, g2), the parity of the signal (S) and the value of the indicator (Ih, Iv); and
- applying (E31) said chosen processing to the digital signal (S) before the spectral recomposition step (E36).

12. Decoding method according to Claim 11 characterised in that said digital signal start and end processings are symmetrical extensions of the digital signal.

13. Decoding method according to one of Claims 9 to 12, characterised in that it also includes a prior step of dividing (E41) the compressed data file (F*) into several digital signals (A, B, C, D) sequenced in a predetermed order, the size of said signals being determined according to the value of the indicators (Ih, Iv).

14. Method of coding a digital signal (S) of dimension N adapted to be transformed geometrically by a transcoding method according to one of Claims 1 to 8, characterised in that it includes the following steps:

- spectral breakdown (E6) into frequency sub-bands of the digital signal (S); and
- recording (E2) in a compressed file (F) comprising the coded digital signal (S) of N indicator or indicators (Ih, Iv) associated respectively with N direction or directions of the digital signal (S) in the form of a supplementary bit having an initial value (0) representing a normal order of the coefficients (c_i) of the frequency sub-bands in a direction associated with said indicator (Ih, Iv).

15. Device for the geometric transcoding of a compressed data file (F) containing a digital signal (S) of dimension N coded by a coding method including at least one step of spectral breakdown (E6) into frequency sub-bands of the digital signal (S), characterised in that it has:

- means of extracting (6) symbols associated with the coefficients (c_i) of the frequency sub-bands in each direction of the digital signal (S);
- means of applying (7) a geometric transformation to said symbols;
- means of updating (8) N indicator or indicators (Ih, Iv) representing a normal or reversed order of the symbols respectively in N direction or directions of the digital signal (S); and
- means of reconstituting (10) the coded digital signal.

16. Transcoding device according to Claim 15, characterised in that the extraction means (10) are adapted to read the coded digital signal, te symbols being entropic codes associated respectively with the coefficients (c_i) of the frequency sub-bands obtained by spectral breakdown.

17. Transcoding device according to Claim 15, characterised in that the extraction means (10) are adapted to perform an entropic decoding of the coded digital signal, the symbols being quantization symbols (q_i) associated respectively with the coefficients (c_i) of the frequency sub-bands obtained by spectral breakdown.

18. Transcoding device according to Claim 15, characterised in that the extraction means (10) are adapted to perform a dequantization of the coded digital signal, the symbols being the dequantized coefficients (c_i) of the frequency sub-bands obtained by spectral breakdown.

19. Transcoding device according to one of Claims 15 to 18, characterised in that the indicator (Ih, Iv) is a supplementary bit recorded in the compressed data file (F), having an initial value (0) representing a normal order of the coefficients of the frequency sub-bands in a direction associated with said indicator (Ih, Iv).

20. Transcoding device according to one of Claims 15 to 19, the digital signal (S) being of dimension 2, characterised in that the means (7) of applying a geometric transformation are adapted to effect a transformation by axial or central symmetry of said symbols, a transformation by rotation through a multiple of 90° or a combination of said transformations.

21. Transcoding device according to Claim 20, characterised in that it also comprises means (9) of transposing a frequency sub-band (HL_3, HL_2, HL_1) having coefficients of low frequency in a first direction of the digital signal and of high frequency in a second direction of the digital signal with a frequency sub-band (LH_3, LH_2, LH_1) with the same resolution level in the spectral breakdown, having coefficients of high frequency in said first direction and of low frequency in said second direction, when the geometric transformation applied comprises a rotation through 90° or 270°.

22. Transcoding device according to one of Claims 15 to 21, the compressed data file (F) containing several digital signals (A, B, C, D) sequenced in a predetermined order, characterised in that it also has means (30) of sequencing said digital signals (A, B, C, D) according to the geometric transformation applied.

23. Transcoding device according to one of Claims 15 to 22, characterised in that the means of extracting (6), applying (7) a geometric transformation, updating (8), reconstituting (10) and if applicable transposing (9) and sequencing (30), are incorporated in:

- a microprocessor (100),
- a read-only memory (102) containing a program for geometrically transcoding a coded digital signal, and
- a random access memory (103) containing registers adapted to record variables modified during the running of said program.

24. Device for decoding a compressed data file (F*) containing a digital signal (S) of dimension N coded by a coding method including at least one step of spectral breakdown (E6) into frequency sub-bands of the digital signal, characterised in that it has:

- means of reading (11) an indicator (Ih, Iv) representing a normal or reversed state of the coefficients (c_i) of the frequency sub-bands in one direction of the digital signal (S);
- means of calculating (14) the parity of the digital signal in said direction;
- means of transforming (18) original spectral recomposition filters (h2, g2) in said direction according to the parity of the digital signal (S) and the value of the indicator (Ih, Iv); and
- means for the spectral recomposition (19) of the digital signal (S*) by means of transformed recomposition filters (h'2, g'2).

25. Decoding device according to Claim 24, characterised in that the means of transforming (18) the original spectral recomposition filters (h2, g2) are adapted to make said filters symmetrical or to shift them by an index.

26. Decoding device according to one of Claims 24 or 25, the method of coding the digital signal (S) comprising, in each direction of said signal, a processing of the start and a processing of the end (E2) of the digital signal (S),

characterised in that it also has:

- means of calculating (15) the parity of the spectral recomposition filters (h2, g2);
- means of choosing (16) the processing of ends to be applied to said digital signal according to the parity of the spectral recomposition filters (h2, g2), the parity of the signal (S) and the value of the indicator (Ih, Iv); and
- means of applying (17) said chosen processing to the digital signal.

27. Decoding device according to Claim 26, characterised in that said processings of the start and end of the digital signal are symmetrical extensions of the digital signal.

28. Decoding device according to one of Claims 24 to 27, characterised in that it also has means of dividing (31) the compressed data file (F*) into several digital signals (A, B, C, D) sequenced in a predetermined order, the size of said signals being determined according to the value of the indicators (Ih, Iv).

29. Decoding device according to one of Claims 24 to 28, characterised in that the means of reading (11), calculating (14) the parity of the digital signal (S), transforming (18), spectral recomposition (19) and if applicable calculating (15) the parity of the filters, choosing, (16), applying, (17) and dividing (31), are incorporated in:

- a microprocessor (100),
- a read-only memory (102) containing a program for decoding the coded digital signal, and
- a random access memory (103) containing registers adapted to record variables modified during the running of said program.

30. Device for coding a digital signal (S) of dimension N adapted to be transformed geometrically by a transcoding method according to one of Claims 1 to 8, characterised in that it has:

- means for spectral breakdown (2) into frequency sub-bands of the digital signal (S); and
- means of recording (5), in a compressed data file (S) containing the coded digital signal, N indicator or indicators (Ih, Iv) associated respectively with N direction or directions of the digital signals (S), in the form of a supplementary bit having an inital value (0) repesenting a normal order of the coefficients (c_i) of the frequency sub-bands in a direction associated with the said indicator (Ih, Iv).

31. Coding device according to Claim 30, characterised in that the breakdown means (2) and the recording means (5) are incorporated in:

- a microprocessor (100),
- a read-only memory (102) containing a program for coding the digital signal (S); and
- a random access memory (103) containing registers adapted to record variables modified during the running of said program.

32. Digital signal processing apparatus, characterised in that it has means adapted to implement the transcoding method according to one of Claims 1 to 8.

33. Digital signal processing apparatus, characterised in that it has means adapted to implement the decoding method according to one of Claims 9 to 13.

34. Digital signal processing apparatus, characterised in that it has means adapted to implement the coding method according to Claim 14.

35. Digital signal processing apparatus, characterised in that it has a transcoding device according to one of Claims 15 to 23.

36. Digital signal processing apparatus, characterised in that it has a decoding device according to one of Claims 24 to 29.

37. Digital signal processing apparatus, characterised in that it has a coding device according to one of Claims 30 or 31.

38. Digital photographic apparatus, characterised in that it has means adapted to implement the transcoding method

according to one of Claims 1 to 8.

39. Digital photographic apparatus, characterised in that it has means adapted to implement the decoding method according to one of Claims 9 to 13.

40. Digital photographic apparatus, characterised in that it has means adapted to implement the coding method according to Claim 14.

41. Digital photographic apparatus, characterised in that it has a transcoding device according to one of Claims 15 to 23.

42. Digital photographic apparatus, characterised in that it has a decoding device according to one of Claims 24 to 29.

43. Digital photographic apparatus, characterised in that it has a coding device according to one of Claims 30 or 31.

44. Digital printer, characterised in that it has means adapted to implement the transcoding method according to one of Claims 1 to 8.

45. Digital printer, characterised in that it has means adapted to implement the decoding method according to one of Claims 9 to 13.

46. Digital printer, characterised in that it has a transcoding device according to one of Claims 15 to 23.

47. Digital printer, characterised in that it has a decoding device according to one of Claims 24 to 29.

48. Computer, characterised in that it has means adapted to implement the transcoding method according to one of Claims 1 to 8.

49. Computer, characterised in that it has means adapted to implement the decoding method according to one of Claims 9 to 13.

50. Computer, characterised in that it has means adapted to implement the coding method according to Claim 14.

51. Computer, characterised in that it has a transcoding device according to one of Claims 15 to 23.

52. Computer, characterised in that it has a decoding device according to one of Claims 24 to 29.

53. Computer, characterised in that it has a coding device according to one of Claims 30 or 31.

54. Photocopier, characterised in that it has means adapted to implement the transcoding method according to one of Claims 1 to 8.

55. Photocopier, characterised in that it has means to implement the decoding method according to one of Claims 9 to 13.

56. Photocopier, characterised in that it has means adapted to implement the coding method according to Claim 14.

57. Photocopier, characterised in that it has a transcoding device according to one of Claims 15 to 23.

58. Photocopier, characterised in that it has a decoding device according to one of Claims 24 to 29.

59. Photocopier, characterised in that it has a coding device according to one of Claims 30 or 31.

60. Computer program product loadable into a computer comprising software code portions for performing the steps of the transcoding method according to one of claims 1 to 8 when it runs on a computer.

61. Computer program product loadable into a computer comprising software code portions for performing the steps of the decoding method according to one of claims 9 to 13 when it runs on a computer.

62. Computer program product loadable into a computer comprising software code portions for performing the steps of the coding method according to claim 14 when it runs on a computer.

63. Computer program product stored on a computer usable medium comprising software code portions for performing the steps of the transcoding method according to one of claims 1 to 8 when it runs on a computer.

64. Computer program product stored on a computer usable medium comprising software code portions for performing the steps of the decoding method according to one of claims 9 to 13 when it runs on a computer.

65. Computer program product stored on a computer usable medium comprising software code portions for performing the steps of the coding method according to claim 14 when it runs on a computer.

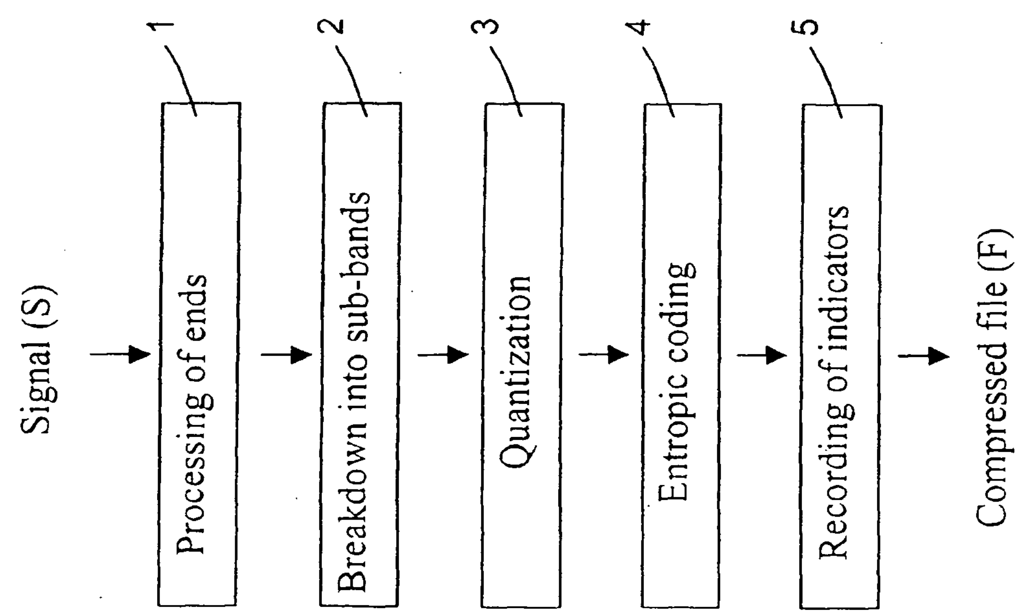

Figure 1

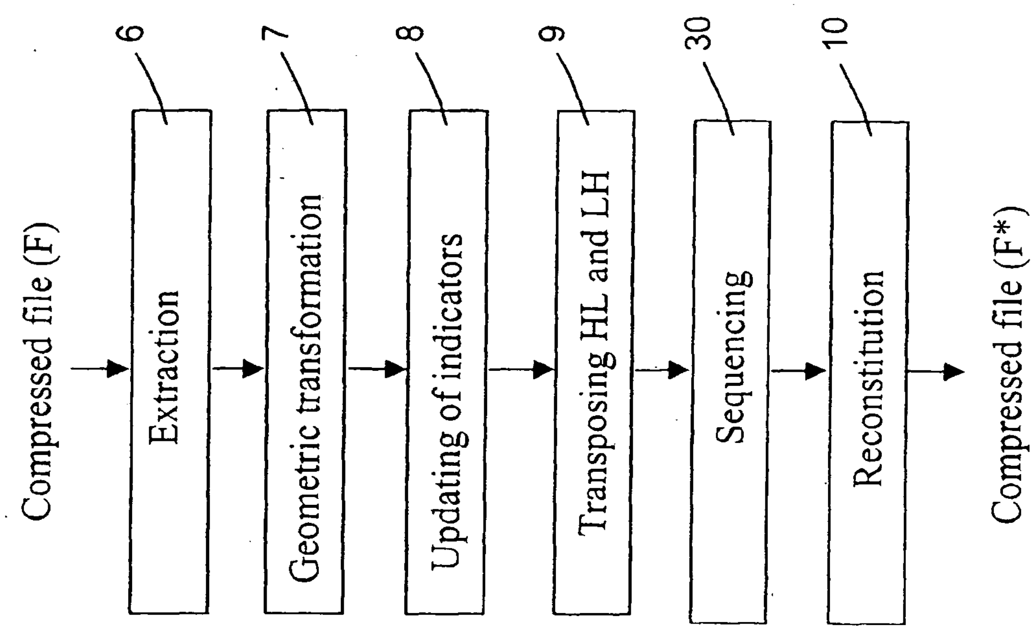

Figure 2

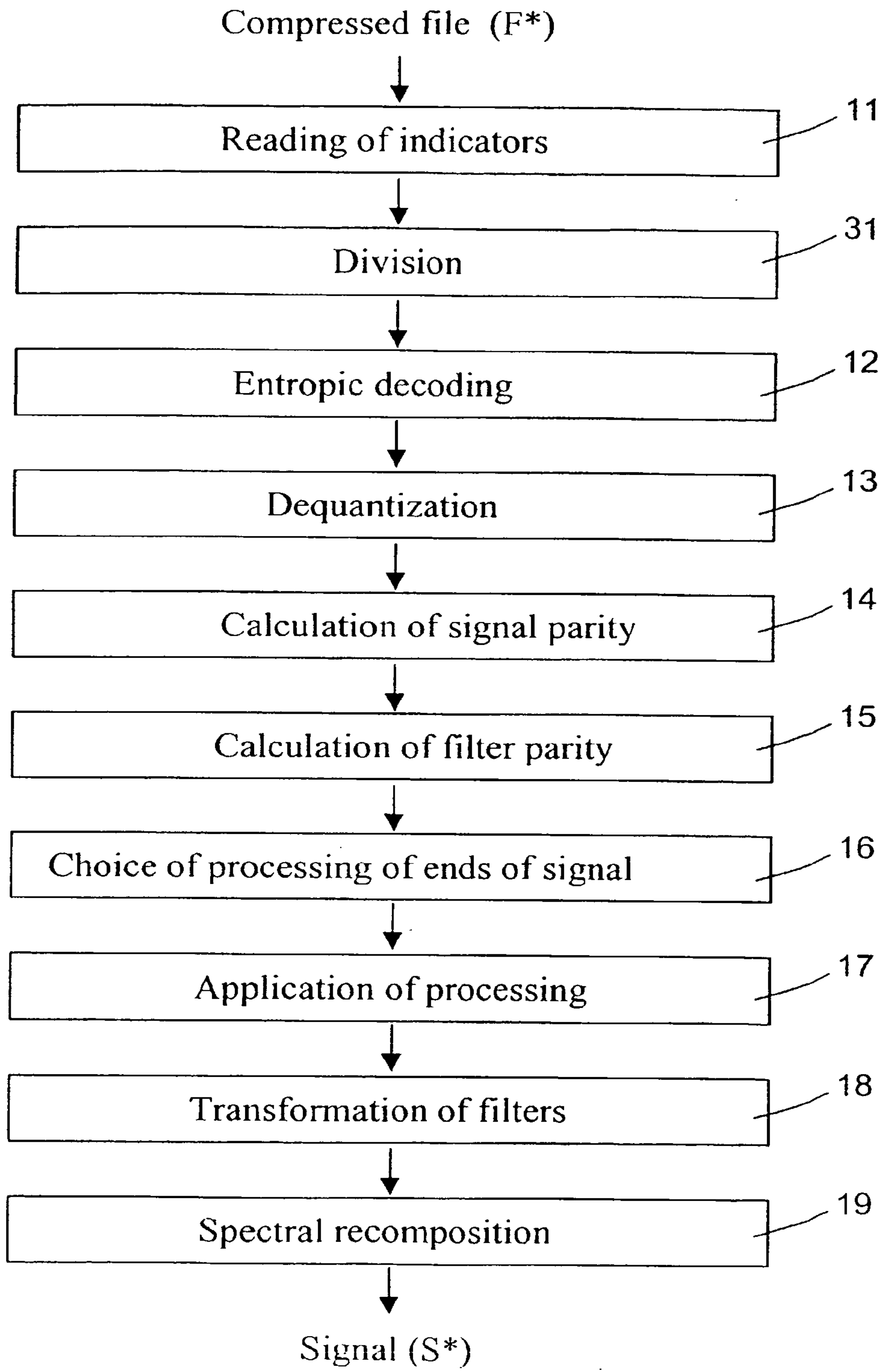
Compressed file (F*)
Reading of indicators
11
Division
31
Entropic decoding
12
Dequantization
13
Calculation of signal parity
14
Calculation of filter parity
15
Choice of processing of ends of signal
16
Application of processing
17
Transformation of filters
18
Spectral recomposition
19
Signal (S*)

Figure 3

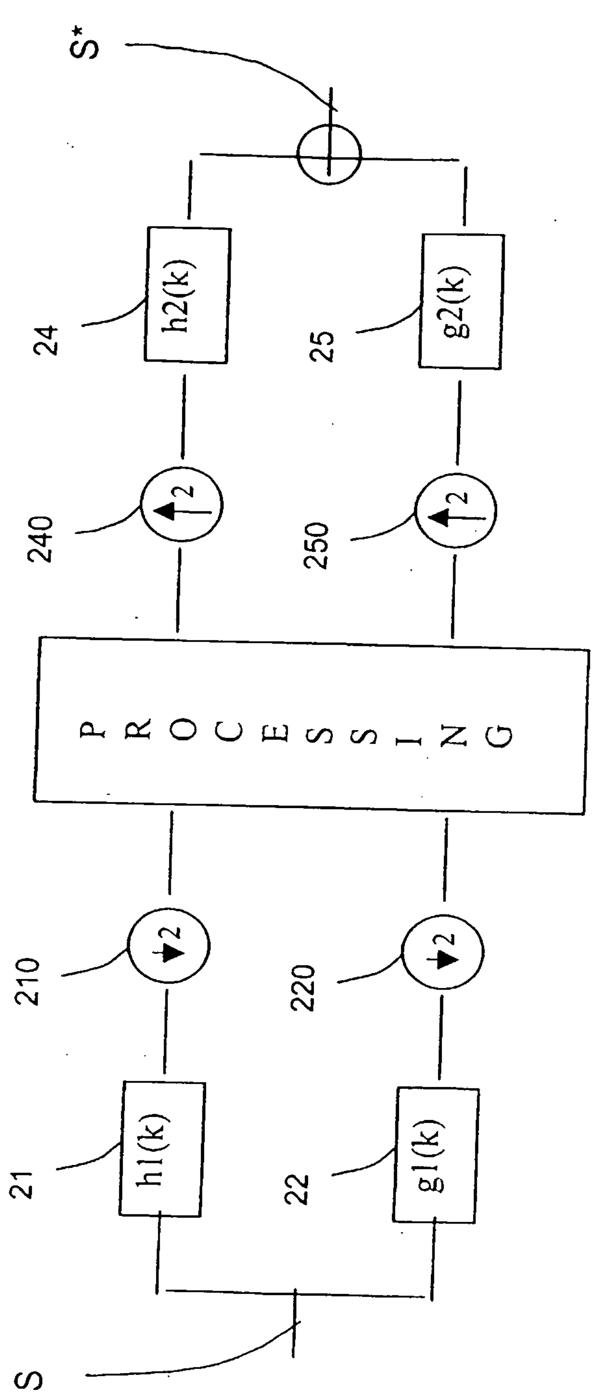

Figure 4

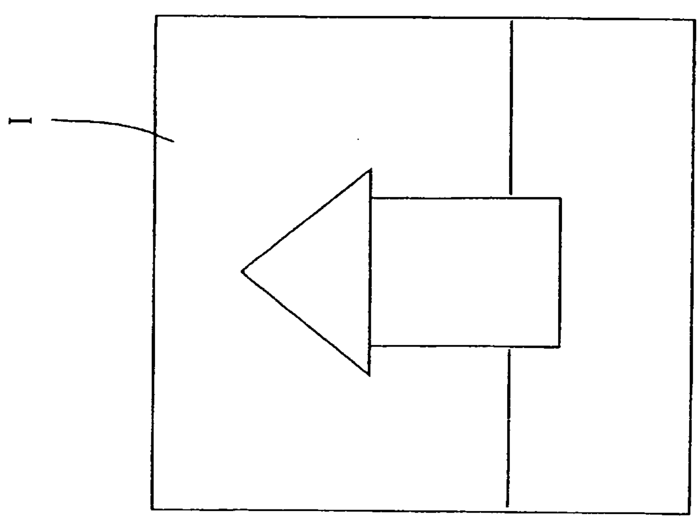

Figure 5

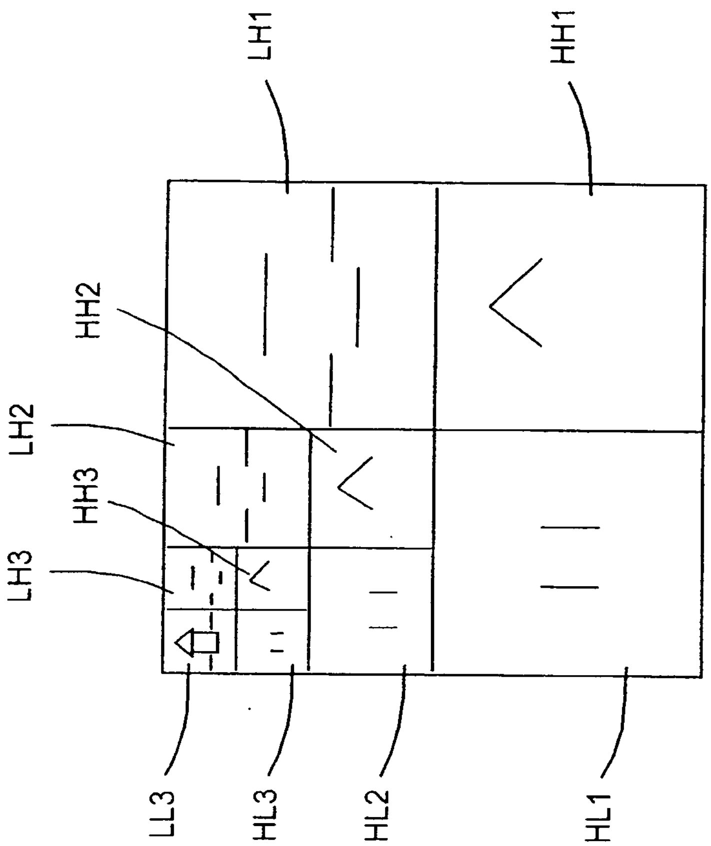

Figure 6

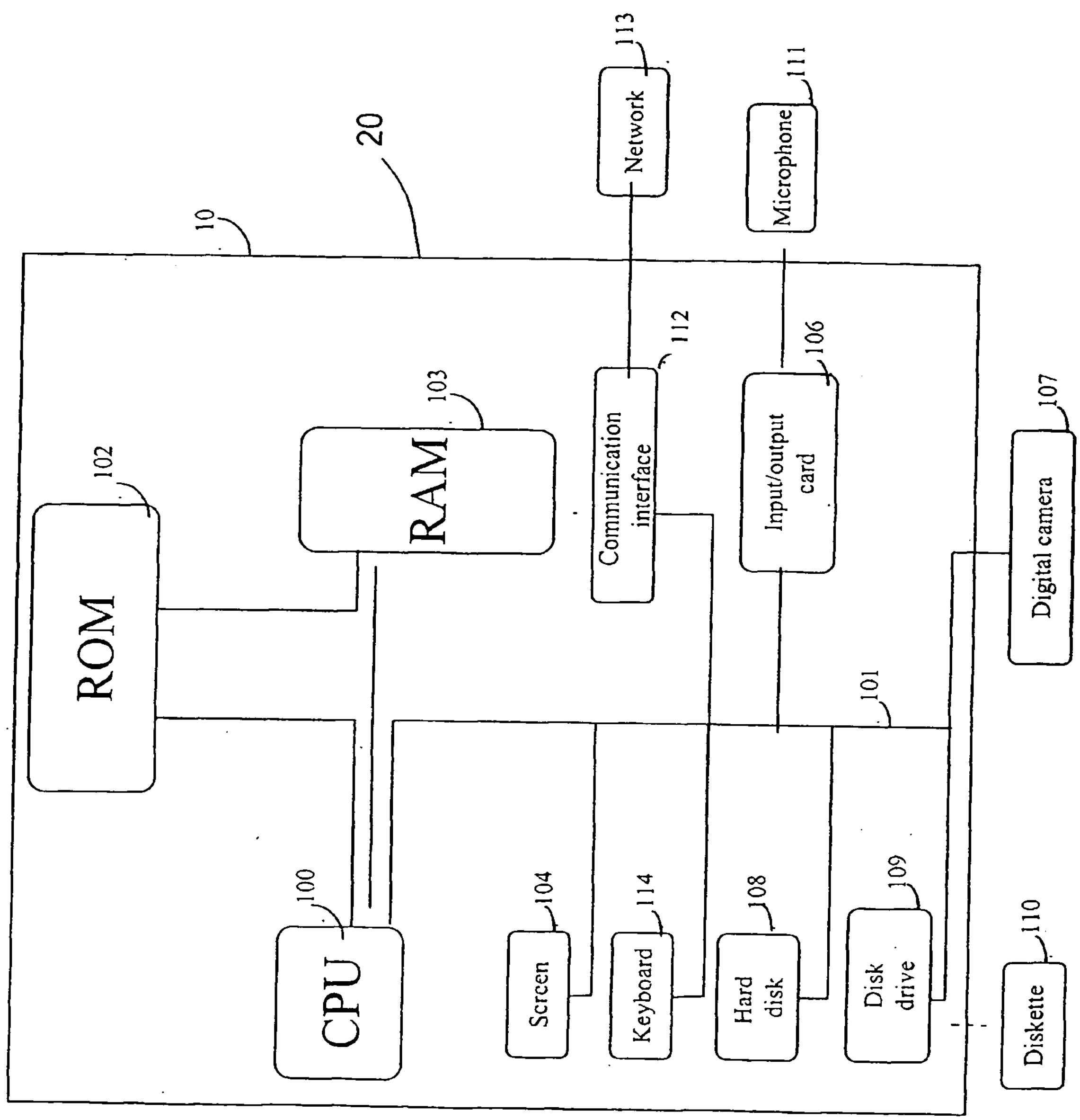
20
10
ROM
102
RAM
103
Communication interface
112
Network
113
Input/output card
106
Microphone
111
Digital camera
107
CPU
100
Screen
104
Keyboard
114
Hard disk
108
Disk drive
109
Diskette
110
101

Figure 7

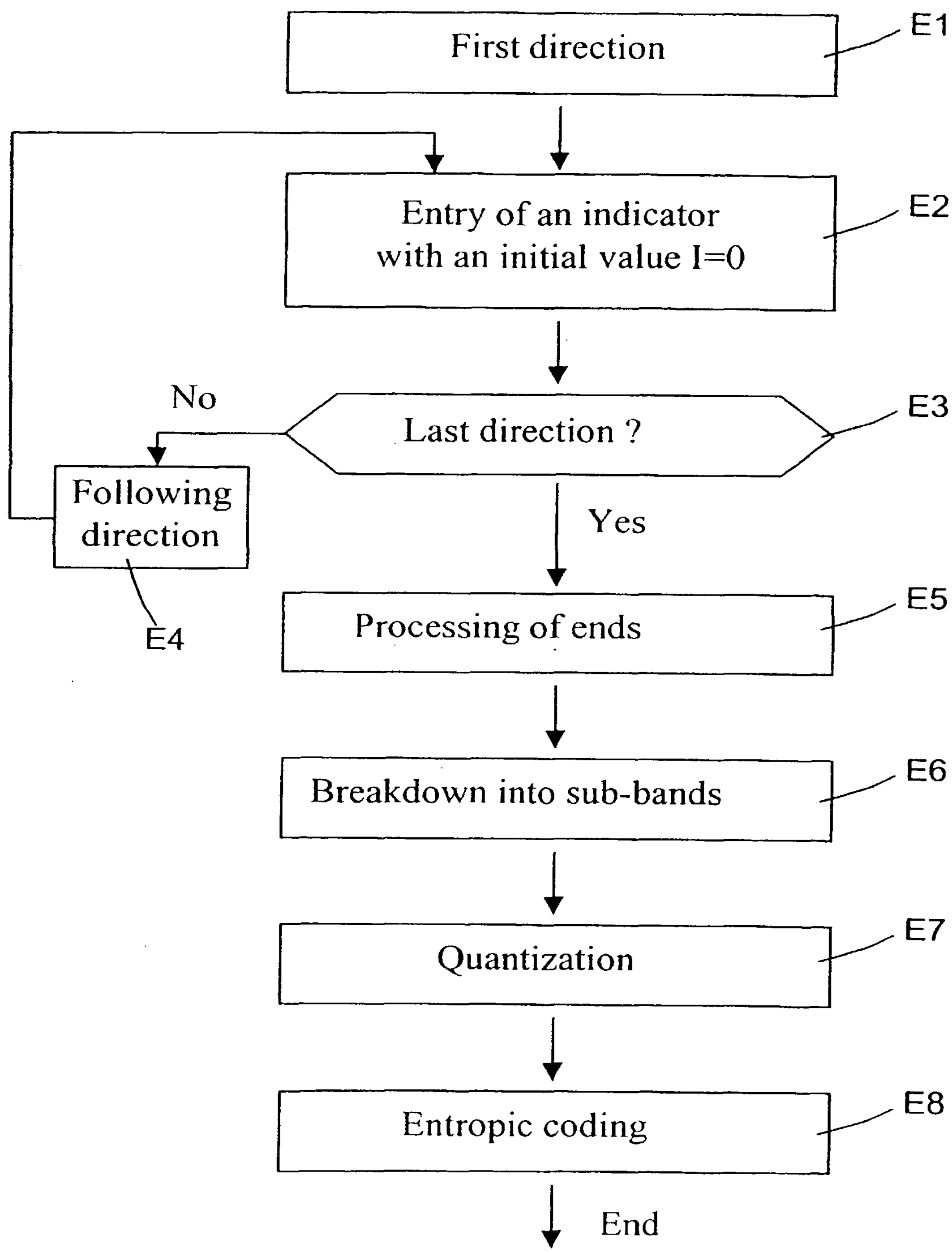
First direction
E1
Entry of an indicator with an initial value I=0
E2
Last direction ?
E3
No
Following direction
E4
Yes
Processing of ends
E5
Breakdown into sub-bands
E6
Quantization
E7
Entropic coding
E8
End

Figure 8

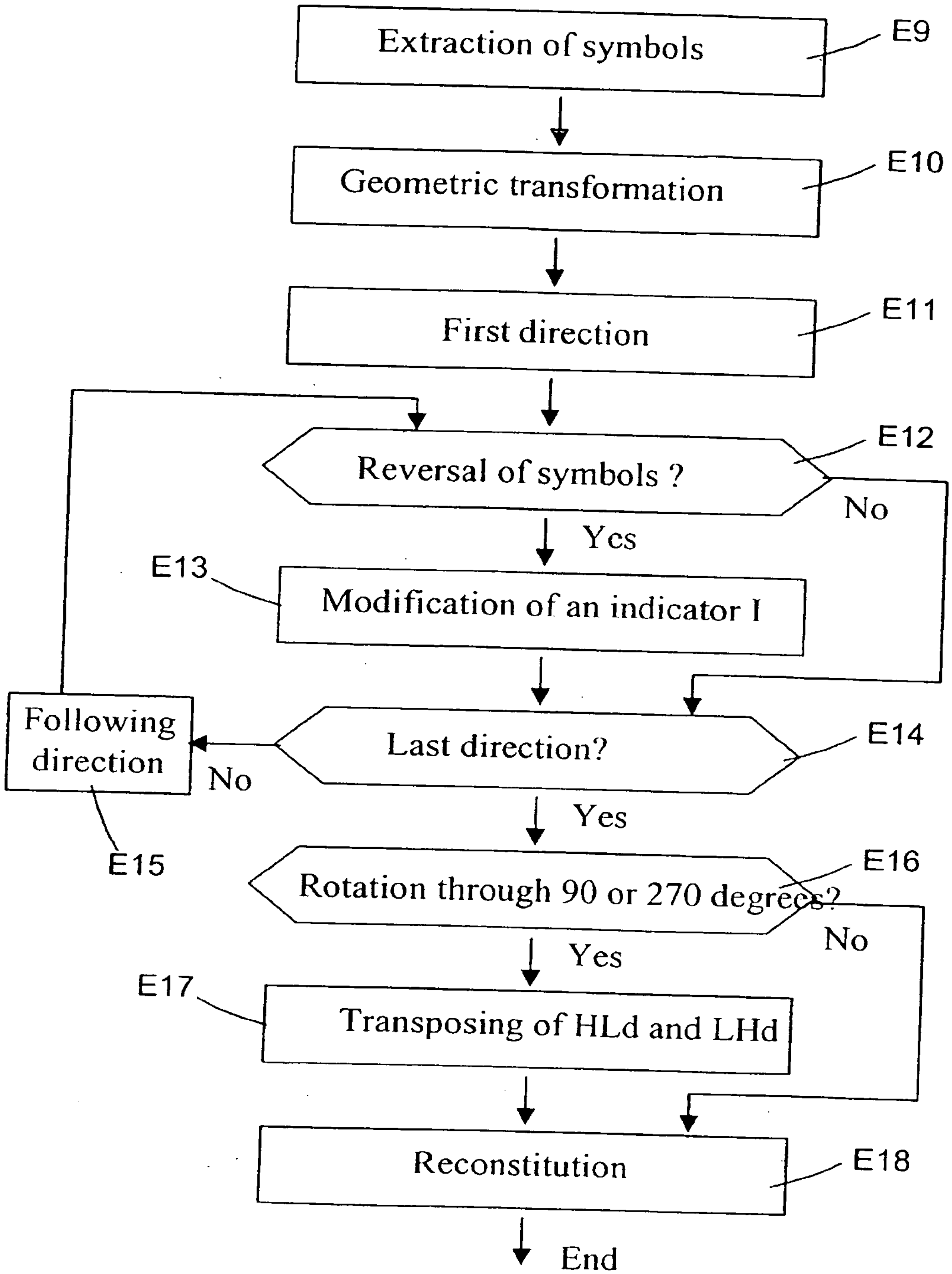
Extraction of symbols
E9
Geometric transformation
E10
First direction
E11
Reversal of symbols ?
E12
No
Yes
E13
Modification of an indicator I
Following direction
Last direction?
E14
No
Yes
E15
Rotation through 90 or 270 degrees?
E16
No
Yes
E17
Transposing of HLd and LHd
Reconstitution
E18
End

Figure 9

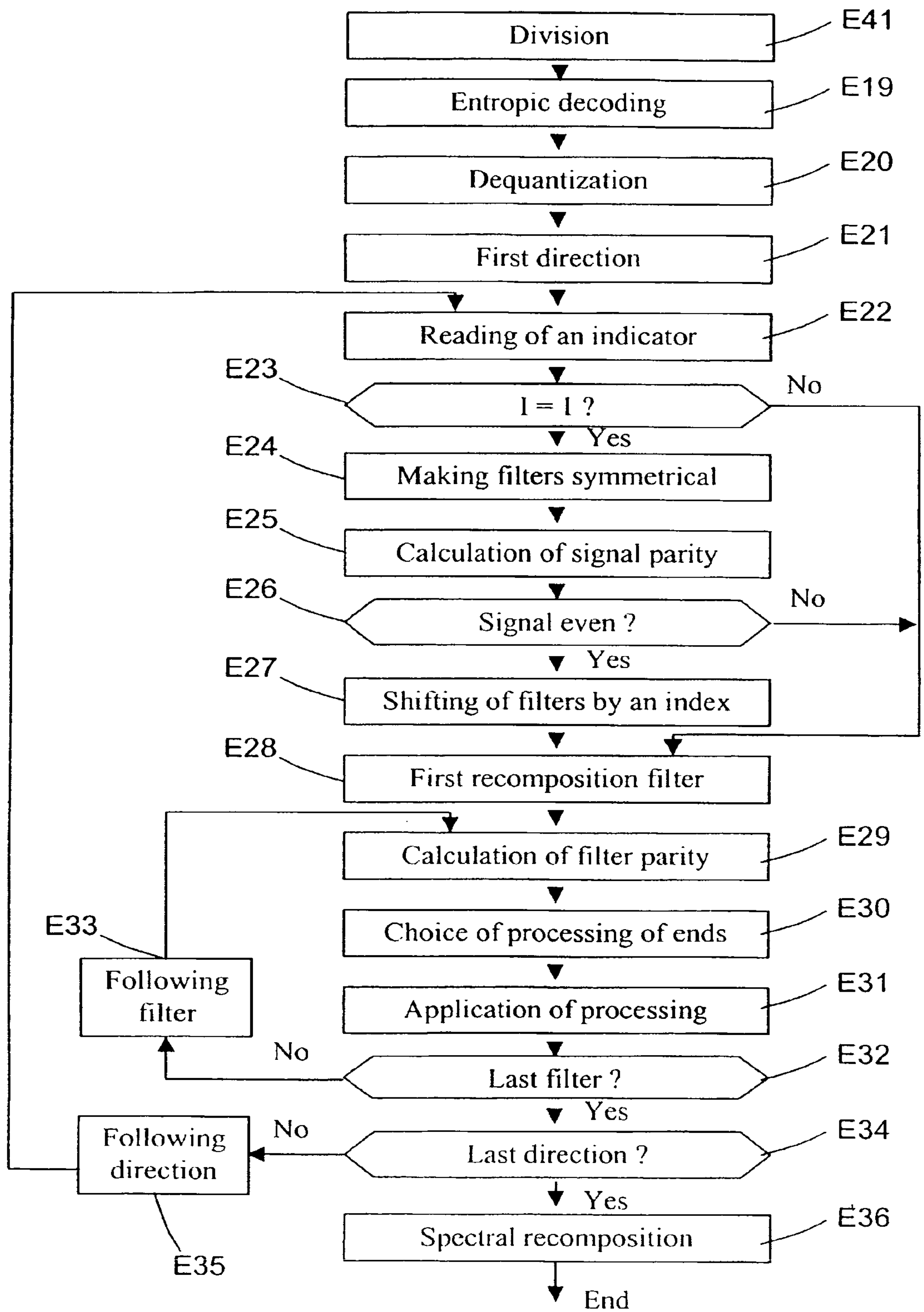
Division
E41
Entropic decoding
E19
Dequantization
E20
First direction
E21
Reading of an indicator
E22
E23
I = 1 ?
No
Yes
E24
Making filters symmetrical
E25
Calculation of signal parity
E26
Signal even ?
No
Yes
E27
Shifting of filters by an index
E28
First recomposition filter
Calculation of filter parity
E29
E33
Choice of processing of ends
E30
Following filter
Application of processing
E31
No
Last filter ?
E32
Yes
Following direction
No
Last direction ?
E34
Yes
Spectral recomposition
E36
E35
End

Figure 10

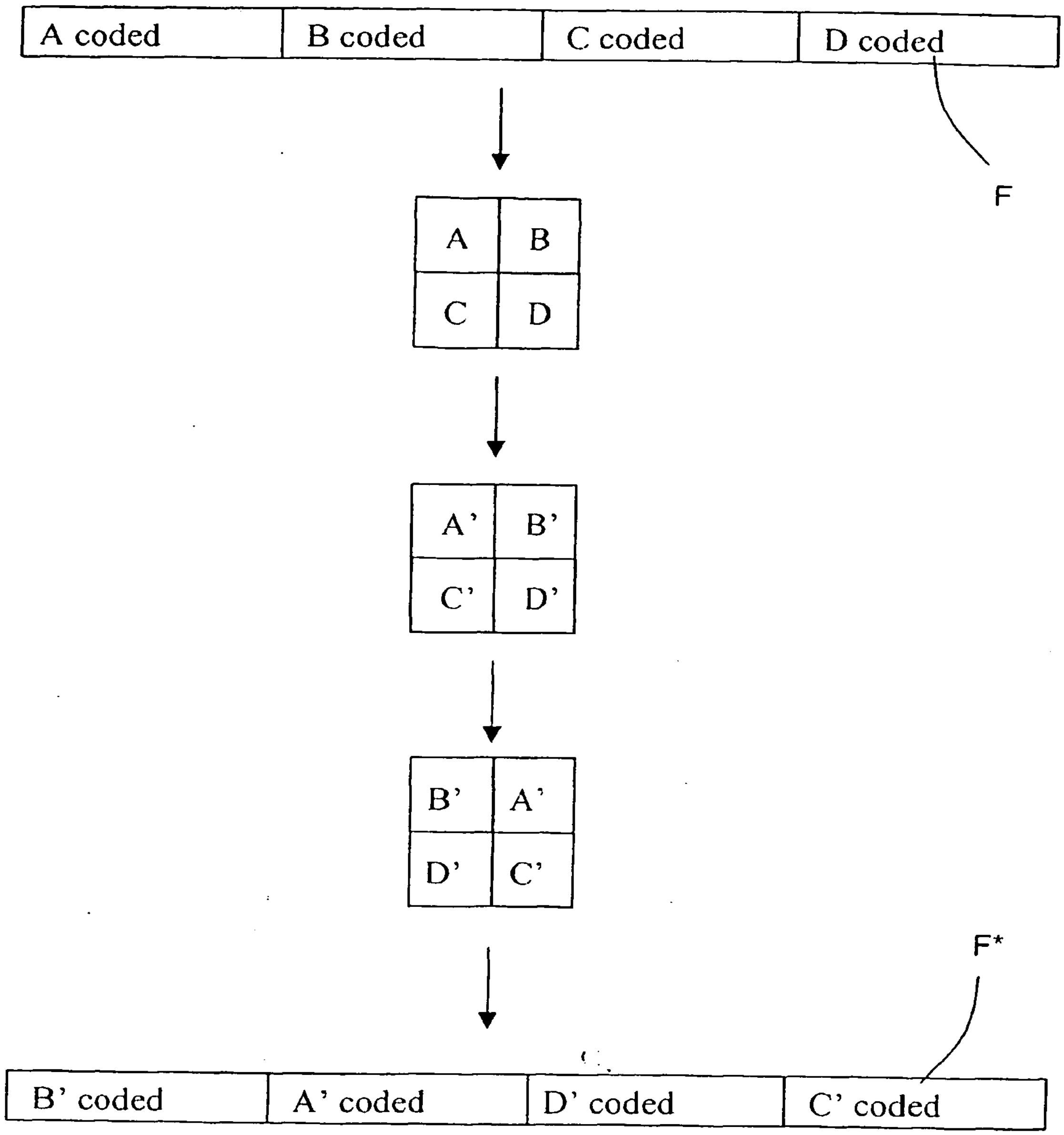
A coded
B coded
C coded
D coded
F
A
B
C
D
A'
B'
C'
D'
B'
A'
D'
C'
F*
B' coded
A' coded
D' coded
C' coded

Figure 11

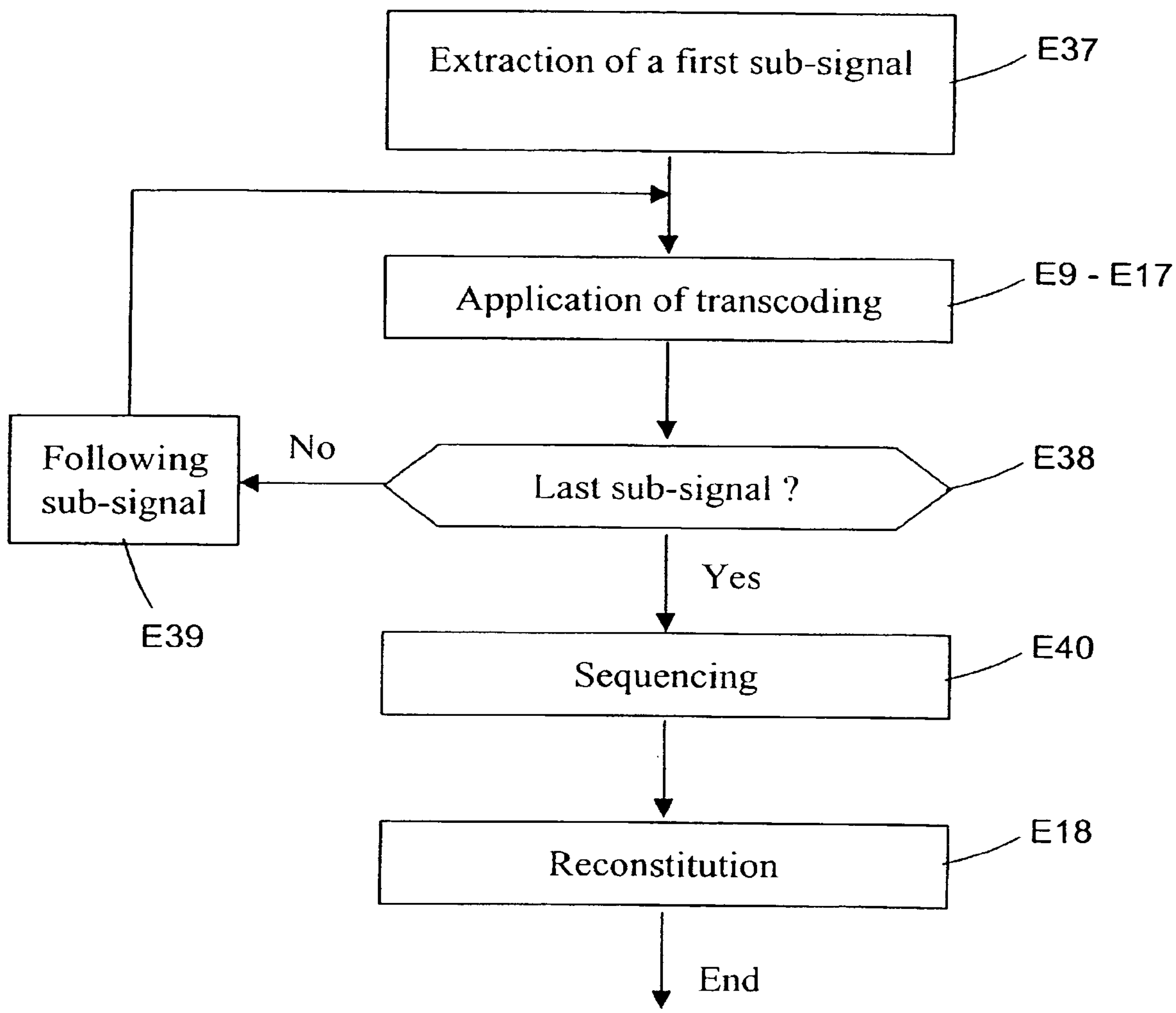
Extraction of a first sub-signal
E37
Application of transcoding
E9 - E17
Last sub-signal ?
E38
No
Following sub-signal
E39
Yes
Sequencing
E40
Reconstitution
E18
End

Figure 12

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 99 40 1970

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	CHANG S -F: "NEW ALGORITHMS FOR PROCESSING IMAGES IN THE TRANSFORM-COMPRESSED DOMAIN" PROCEEDINGS OF THE SPIE, vol. 2501, no. PART 01, 24 May 1995 (1995-05-24), pages 445-454, XP000610425 * paragraph '0001!; figure 1 * ---	1,9,14, 15,24, 30,32-65	H04N1/41
A	LHUILLIER J J ET AL: "SUBBAND CODING OF IMAGES - COMPARISON WITH DCT" SIGNAL PROCESSING: THEORIES AND APPLICATIONS, GRENOBLE, SEPT. 5 - 8, 1988, vol. 3, no. CONF. 4, 5 September 1988 (1988-09-05), pages 1645-1648, XP000186364 LACOUME J L;CHEHIKIAN A; MARTIN N; MALBOS J * paragraph '0001! - paragraph '0002! * ---	1,9,14, 15	
A	SMITH B C ET AL: "COMPRESSED DOMAIN PROCESSING OF JPEG-ENCODED IMAGES" REAL-TIME IMAGING, vol. 2, no. 1, 1 February 1996 (1996-02-01), pages 3-17, XP000656168 ---		TECHNICAL FIELDS SEARCHED (Int.Cl.7) H04N H03M
A	SMITH B C ET AL: "ALGORITHMS FOR MANIPULATING COMPRESSED IMAGES" IEEE COMPUTER GRAPHICS AND APPLICATIONS, vol. 13, no. 5, 1 September 1993 (1993-09-01), pages 34-42, XP000562744 * paragraphe intitulé "Compressed domain processing" * -----		

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	14 December 1999	Augarde, E

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)